INDEKS nr 32034 X ISSN 1230-7947

Commodore

4/1994

AMIGOWIEC

Pismo użytkowników komputerów AMIGA cena 25.000 zł

SUPERMEMO

Oprogramowanie:

- REAL 3D v2.47
- TURBO CALC
- VIDEO TRACER
- AMOS PRO
- VOYAGER

SPRZĘT:

- PCMCIA
- RAM

WARSZTATY!

- AMOS
- Amiga DOS cz.1
- Arexx cz. 5
- Imagine cz. 4
- Amiga i muzyka
- Modemowanie

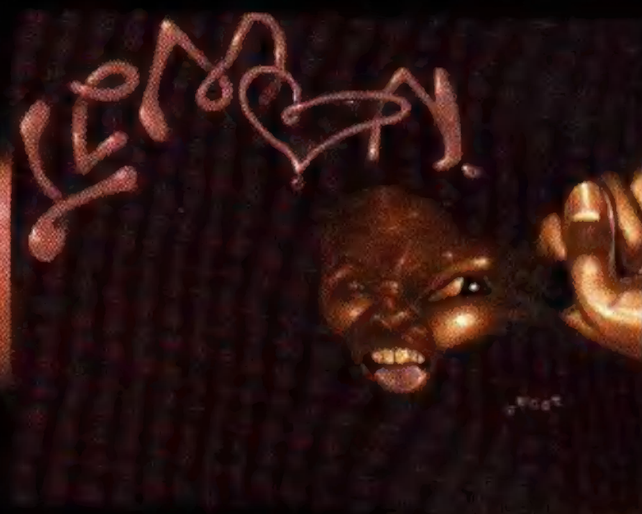
ARAM
Warszawa
tel. (0-22) 29-54-99
DEALER
GVP



3M










R. W. O.





AMIGOWIEC

ROK 5 NR 4/94 (38) INDEX 32034X
© Copyright by P.W.H. "ALFIN" sp. z o.o.

REDAKCJA

REDAKTOR NACZELNY:
RYSZARD KOWALSKI

Z-CA REDAKTORA NACZELNEGO:
KRZYSZTOF NOWICKI

SEKRETARZ REDAKCJI:
TOMASZ ŁOBODA

OPRACOWANIE GRAFICZNE:
KRZYSZTOF WIRSZYŁŁO
BARTŁOMIEJ WRÓBLEWSKI

SKŁAD KOMPUTEROWY:
MAŁGORZATA LEWANDOWSKA

KOREKTA
KAROL JAGODZIŃSKI
ŁUCJA UMIŃSKA

KOLPORTAŻ:
ANDRZEJ KENTZER

PRENUMERATA:
AGNIESZKA PRZYBYLSKA

WSPÓŁPRACA:
WOJCIECH BIAŁKOWSKI, WOJCIECH CZYŻ, JAROSŁAW CHROSTOWSKI, MARCIN GACKOWSKI, ADAM GREGROWICZ, TOMASZ FLANC, TOMASZ HRYCUNIAK, MACIEJ KLIMKIEWICZ, TOMASZ KOKOSZCZYŃSKI, SEBASTIAN KŁOMSKI, TOMASZ KULBACKI, TOMASZ ŁOBODA, ARTUR ŁUKASIK, TOMASZ MATAJEK, ROBERT RUTKOWSKI, IZABELA SKIBIŃSKA, MAREK STOR, KRZYSZTOF TROJANOWSKI,

ADRES REDAKCJI:
ul. Świętojańska 2/7
85-017 Bydgoszcz
tel. (+52) 28-79-20, fax (+52) 22-64-03

KONTO BANK PKO S.A. BYDGOSZCZ
00509011-04009339-2511-30-00111o

SKŁAD NA KOMPUTERACH AMIGA
OKŁADKA: WOJCIECH BIAŁKOWSKI
DRUK: "ZAKŁADY GRAFICZNE" SP. Z O.O.
UL. OKRZEI 5, 64-920 PIŁA
Materiałów niezamówionych redakcja nie zwraca.
Redakcja nie odpowiada za treść ogłoszeń.
COMMODORE *AMIGA* są zastrzeżonymi znakami firmy COMMODORE.

Niespełniony boom



Przed targami CeBit '94 krążyły różne pogłoski. Jedne mówiły o nowych komputerach, inne o upadku firmy. Analizując wiadomości z targów w Hannoverze, można po raz kolejny stwierdzić, że pogłoski to zazwyczaj pogłoski. Commodore nie pokazał żadnego nowego komputera i skupił się na promowaniu już dostępnych modeli. Z "nowych" rzeczy na uwagę zasługuje fakt oficjalnego wprowadzenia do sprzedaży pokazanej rok temu A4000 Tower,napędu CD-ROM dla A1200 oraz promocja modułu MPEG (FVM) przeznaczonego dla CD32, dla którego dostępnych jest już kilka tytułów m.in. z filmów Czas Patriotów (Harrison Ford), Top Gun (Tom Cruise), z koncertów Bon Jovi. W zapowiedziach znajdujemy m.in. filmy Firma (Tom Cruise), Polowanie na Czerwony Październik, Sliver (Sharon Stone). Troszeczkę martwi zbyt wolno powiększająca się liczba gier dla CD32, przynajmniej w Niemczech. Tytuły najnowszych gier pojawiają się tam z dużym opóźnieniem. Podobnie rzecz się ma z firmą GVP, której nowe produkty widnieją na razie tylko w ogłoszeniach. Tak więc spodziewany boom nowych produktów nie nastąpił. Obserwujemy równomierny rozwój oprogramowania i sprzętu prawdopodobnie zgodny z planami Commodore.

Jedną rzecz trzeba stwierdzić z całą opowiedzialnością, ostatnie działania Commodore, jakkolwiek by się o nich wyrażać, doprowadziły do powrotu na rynek Amigowski paru ważnych firm, np. Psygnosis, która w 1993 pożegnała się z Amigą definitywnie, teraz wypuściła nowe tytuły jak Perihelion, Armour Geddon 2. Tak samo powracająca Sierra zadebiutowała programem King's Quest 6. Nieźle, co?

Zeszłoroczne zmiany w polityce Commodora przyniosły owoc nie tylko w postaci gier, ale i sporej grupy profesjonalnych programów użytkowych utrzymujących, tak do tej pory gwałconą przez domorosłych programistów, linię programowania zgodną z systemem operacyjnym. Final Writer, WordWorth 3, Turbo Calc, Superbase Professional 4.x, Pro Vector 3, Page Stream 3, Envoy (system sieciowy), Real 2.x, to cała gama oprogramowania zgodnego z linią systemu.

Wygląda na to, że Commodore chce pójść za ciosem. Na ukończeniu znajdują się prace nad OS 3.1, który ma być ostatnim systemem operacyjnym przyznaczonym dla dotychczas wyprodukowanych modeli, niejako łączącym je i wyrównującym możliwości programowe. Chodzą słuchy, że system będzie posiadał poprawione sterowniki obrazu, wbudowany system obsługi CD-ROM zgodny ze standardem CD32 oraz wreszcie system RTG, który uporządkuje standard kart graficznych dla Amigi.

W tym roku ma zostać także wprowadzony nowy komputer. Kiedy? Czas pokaże. Jednak z przecieków oraz wypowiedzi inżynierów Commodore udało się nam złożyć pewną ogólną wizję możliwości nowego komputera, którą przedstawimy w następnym numerze Amigowca.

Co zaś się tyczy naszego rynku, dalej pozostaje on nieuporządkowany. Ostatnio doszły nas słuchy, że w niektórych miastach Polski pojawiły się Amigi z Włoch. Dobre, co? Commodore, oj Commodore, chciałoby się zawyć.

W imieniu naszego wydawnictwa pragnę zaprosić wszystkich na targi Infosystem '94, gdzie będziemy obecni na stoisku w hali 26 poziom B (parter) w dniach 17-20.05.94. Jedną z atrakcji będzie prezentacja roboczej wersji programu Page Stream 3.

Dziękując za bardzo spontaniczne potraktownie naszej Ankiety oraz ogromne ilości listów, które napływają do naszej redakcji, a na które nie jesteśmy w stanie odpowiadać indywidualnie, zapraszam do lektury kolejnego numeru Amigowca.

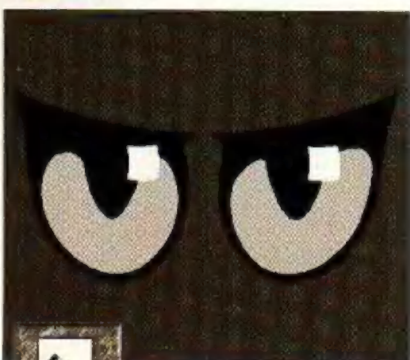
R. Kowalski





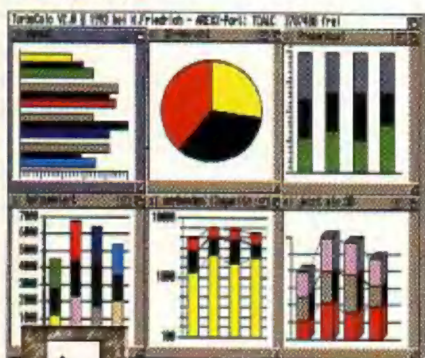
10 PCIMCIA

Przegląd rodzajów kart PCMCIA na który zaprasza Tomek Kulbacki




11 SUPER MEMO

Rewelacyjna nowa metoda uczenia się. Za sprawą firmy TSS Super Memo również w wersji na Amigę.




14 TURBO CALC

Firma Stefan Ossowski Schatzruhe dostarczyła nam nowy arkusz kalkulacyjny Turbo Calc. Przyjrzał się mu Tomasz Kokoszczyński.




18 AMOS PRO

Tomak Hrycuniak przedstawia pakiet Amos Professional.


EdWord Pro Text Editor








TRYBUNA




Amiga
DOS







Zapraszamy na Targi!

Wydawnictwo ALFIN ma zaszczyt zaprosić Państwa na Targi Infosystem '94 w dniach 17-20.05.94 do stoiska w hali 26 poziom B (parter). Na stoisku będą obecni członkowie redakcji Amigowca, którzy zaprezentują najnowsze wersje programów DTP na Amigę: Page Streama 3, Type Smitha 2 oraz Art Expression, a także członkowie redakcji Świata Gier Komputerowych. Firma przewiduje sprzedaż tanich rozszerzeń pamięci firmy MTEC do A1200. Zapraszamy!

Czcionka na zamówienie

Wszyscy wiemy jak trudno czasami jest dobrać czcionkę, która jest nam akurat potrzebna. A to krój nie ten, a to wielkość nie ta. A gdy w końcu dopadniemy czcionkę naszych marzeń to okazuje się, że nikt nie wpadł na to żeby do niej dorobić polskie literki. Wprawdzie istnieją edytory czcionek jak stary dobry FED, Caligrapher, czy ostatnio TypeSmith 2.0 umożliwiający edycję czcionek bitmapowych. Jednak przy pewnych zastosowaniach to ciągle jest nie to. A co jeśli zamarzy nam się czcionka kolorowa? Np. złota? Caligrapher wprawdzie umożliwia tworzenie takich czcionek, ale współpraca z nim to raczej katorga niż radość tworzenia nowego kroju (bardziej zagmatwaną obsługę chyba jest trudno wymyślić).

Wszystkim zmartwionym z odsieczą przychodzi program Personal Fonts Maker 2.0., stworzony przez Cloanto (twórców Personal Painta). Program ten pozwala na wyjątkowo proste tworzenie efektownie wyglądających czcionek, które z powodzeniem można stosować w tytułach w naszym domowym Video (przecież z komputera taki tytuł bez problemu da się wstawić na kasetę), w zastosowaniach multimedialnych itp.

Obsługa programu jest podobna do Personal Painta i jego użytkownicy nie powinni mieć z nią żadnych problemów. Program współdziała z pełną paletą 256 kolorów kości AGA i jest wyposażony w opcje anti-aliasingu oraz wykorzystywania czcionek wektorowych "na podkładkę". Jego cena wynosi ok. 60 funtów.

Amigowe podziały wieku

Amigi są, jak wszyscy wiemy, bardzo popularne w świecie gier. W przerwie między jedną amigową pracą a drugą, każdy rasowy amigowiec chętnie pogra. Gry nie są tak nieszkodliwe jak programy użytkowe i nie wszyscy mogą ich używać. Do takiego słusznego wniosku doszli niektórzy prawnicy w Anglii. Po burzliwej nagonce na gry wideo od marca na wszystkich grach komputerowych mają się pojawić nalepki mówiące dla jakiego zakresu wiekowego są dane gra. Przedziały są następujące: 0 - 10, 11 - 14, 15 - 17 i 18 +. Indywidualni wydawcy gier zostali zobowiązani do współpracy z Radą Standardów Video (Video Standard Council) w zakresie ustalania haczyków na nalepkach. Chodzi o to, żeby nalepki były zgodne z angielską klasyfikacją filmów. Przypuszcza się, że tylko 5% produktów wypadnie z tych ramek.

Nalepki są bardziej przeznaczone dla rodziców niż dla dzieci. Nasze pociechy najchętniej grałyby we wszystko. Pamiętajmy jednak o tym, że zakazany owoc lepiej smakuje i skuteczniejszym sposobem dla zniechęcenia dzieciaka jest przetrzymanie go przy 2 godzinnym melodramacie "tylko dla dorosłych" niż tłumaczenie mu, że to nie dla niego i powinien iść odrabiać lekcje. Dzieci doskonale wyczuwają intencje dorosłych. Głupie przecież nie są. Z grami oczywiście bywa różnie i sposób ten nie jest zawsze skuteczny. Dlatego proponuję, żeby wprowadzić specjalne nalepki "Tylko dla dorosłych", które zostałyby umieszczone na wszystkich poważnych programach użytkowych. Założę się, że znakomita większość nastolatków zniechęci się do CADa po kilku minutach, jeśli nie szybciej...

Trudny trójwymiar

Kalifornijska firma Anti Gravity Products zakończyła prace nad grupą programów umożliwiających tworzenie naturalnie wyglądających modeli trudnych obiektów 3D, używanych z takim umiłowaniem w programach typu Real 3D, czy Imagine.

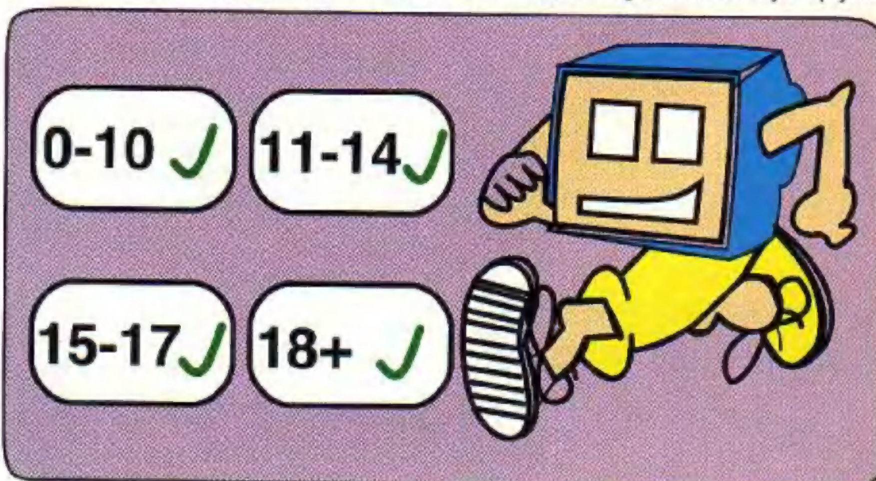
Pierwszy z nich to Snap Maps. Jest to program do łączenia płaskich tekstur i innych wzorców rozrysowanych w 24-bitowej grafice. Program ten pozwala na korzystanie z matryc dostarczanych z pakietami 3D do tworzenia wyrazistych grafik.

Drugą częścią pakietu jest Fields and Foliae, zawierający wytyczne do tworzenia płatków, winorośli, kory, gałęzi, liści, trawy i wielu podobnych "niezgrabnych" obiektów.

Zestaw Materials and Fabrics zawiera plecionki, materiały, siatki, a nawet płoty.

HELM wchodzi do akcji

HELM to nowy system multimedialnej obsługi Amigi stworzony przez firmę Eagle Tree Software. Jest to coś takiego jak Amiga Vision, dostarczany swego czasu z Amigami 3000, czy w mniejszym zakresie Scala. Program został wyposażony w nowoczesny edytor akcji, którego zadaniem jest płynne





i pełne łączenie grafik, animacji, tekstów, dźwięków i plików muzycznych ze sobą tak, aby w trakcie prezentacji wszystkim "gały wyszły na wierzch". Pragnę tu zwrócić Waszą uwagę, że jest to program do sterowania całością prezentowanego materiału, a nie do jego tworzenia. Program jest w pełni kompatybilny z układami AGA. Jego rozprowadzaniem zajmuje się angielska firma Meridian Software Distribution i kosztuje jedyne 100 funtów.

Vista na "małe" Amigi

Jak niektórzy z Was zdążyli się zapewne zorientować, VistaPro jest programem wymagającym minimum 4 MB i koprocesora. Nie każdy może sobie pozwolić na taki zestaw. Wszystkich tych, którzy mają więcej ambicji niż RAMu ucieszy nowina, że powstał program VistaPro Lite. Właśnie dla nich. Program nie został zubożony i krajobrazy, które z niego wyjdą wcale nie muszą być gorsze od tych wytworzonych przy pomocy "prawdziwej" Visty. Stało się to za sprawą nowych technik pamięciooszczędnych wykorzystanych do generowania gór, lasów, chmur i jezior. Tylko lata czekać...

Sprzętowe zabezpieczenie przed wirusami

Firma Silica System wprowadziła na rynek nowe stacje dysków Amitek's Deluxe. Stacje te zostały wyposażone w technologię anty-klikową i sprzętowe zabezpieczenie przed wirusami. Mechanizm obracający dyskietkę oparty jest na technologii Sony. O ile z klikaniem stacji posiadacze systemów od 2.0+ w górę radzą sobie całkiem nieźle dzięki programikowi NoClick, o tyle gorzej z wirusami. Stacja zabezpiecza tylko przed wirusami zapisanymi na ścieżce zerowej, a to zadanie akurat doskonale spełnia VirusChecker. Na dodatek chroni on nas przed większością wirusów plikowych (o ile mamy nową wersję). Żadna stacja nie jest w stanie uchronić nas przed takimi wirusami. Mogłaby jedynie zapobiec w pewnym stopniu ich rozprzestrzenianiu się, ale to sprawa przyszłości. Cacko kosztuje jedyne 59 funtów.



Cyfrowi ludzie w cyfrowym świecie

Praktycznie na co dzień stykamy się z produktami programów 3D. A to pojawi się coś nowego w telewizyjnej reklamówce, a to obejrzymy sobie kawałek figurki stworzonej w Imaginie, czy ostatnio modnym Realu 3D. Trójwymiarowa rzeczywistość staje się faktem (jakby od stuleci było inaczej). Pewnie tylko niewielka część z Was zdaje sobie sprawę jak trudno jest graficznie odwzorować rzeczywisty świat. To dlatego na różnego rodzaju dyskach są rozprowadzane skomplikowane figury statków kosmicznych, butów, czy ludzkich twarzy.



Ostatnio pojawił się na rynku profesjonalnie przygotowany pakiet o dźwięcznej nazwie Humanoid. Figury jak figury - gotowe do przetwarzania w programach Lightwave i Imagine. Tylko, że te dokładnie odwzorowane "obiekty" to dopiero początek tworzenia animowanych filmów z życia cyfrowych lalek. Oprócz tego do pakietu dołączono liczne sceny mówienia, biegania i chodzenia. Te figury po prostu żyją. Dzięki hierarchicznemu podzieleniu ich na poszczególne części ciała i nadaniu im odpowiedniego zakresu ruchów. Jak wspomniałem figury mają także zdolność bardzo realistycznego mówienia. Stało się to możliwe dzięki dokładnemu odtworzeniu wyglądu głów wypowiadających poszczególne fonemy (głoski). Teraz wystarczy tylko "przeprowadzić" głowę z jednego fonemu do następnego i wszystko "gra". Podobnie postąpiono z emocjami. W ten sposób figurki mogą być smutne, złe, wesołe itp. Tim Wilson autor pakietu podobnie doskonale odtworzył ruchy ludzkiej ręki.

W skład zestawu wchodzi cała rodzina (dziecka możemy się spodziewać dopiero po wysłaniu karty rejestracyjnej). Na dodatek wszystkie figury składają się z tej samej liczby punktów tak, że przejścia taty w mamę i na odwrót nie są żadnym problemem.

Dla ostudzenia Waszych zapałów dodam, a figurki ledwo działają z 10MB pamięci RAM i 19MB pamięci dyskowej, że o procesorze 030 lub jeszcze lepiej 040 nie wspomnę.

Cena pakietu wynosi ok. 135 funtów, a wyprodukowała go firma Crestine Software.

Zewnętrzny dysk dla A1200

Wszyscy pamiętamy stare dobre czasy, gdy ktoś ze znajomych podłączał do swojej A500 twardziela. Oczywiście twardziel ten był umieszczany z boku i dołączało się go do szyny Amigi. Ta możliwość przestała być aktualna z chwilą pojawienia się Amig 600 i 1200. One po prostu nie mają takiej szyny.



Oczywiście rynek nie znosi próżni i pojawiły się na nim dyski, które można podłączyć do portu PCMCIA. Jednym z nich jest Overdrive. Ostatnio wstawiono do niego nowy kontroler i dołożono własne zasilanie tak, żeby nasze przyjaciółki zbytnio się nie męczyły. Dysk jest sprzedawany w pojemnościach 250 i 340MB, a szybkość transferu danych wynosi od 1.5 do 2.5MB na sekundę i jest o wiele większa niż szybkości wewnętrznych dysków 2.5. Ceny dysku wynoszą odpowiednio 300 i 380 funtów.

Nadchodzi rewelacyjny pakiet graficzny

Nikt jeszcze nie wie jak się będzie nazywał. Praktycznie poza programistami nikt nie zna jego możliwości. Chodzą tylko plotki, że będzie to rewolucja. Pakiet ma się szczególnie nadawać dla potrzeb telewizji i profesjonalnych grafików.







Rewolucja ma polegać na możliwości tworzenia grafik zamiast z wielu warstw "szkła" na jednej płaszczyźnie. Było to w pewnym stopniu możliwe nawet w DeLuxe Paintcie, ale teraz ma to być podstawą działania programu. Do tej pory takie systemy kosztowały ok 150 tysięcy dolarów. W tej chwili ma wystarczyć 32 bitowa Amiga. Program jest tworzony w Amiga Centre Scotland, a jego końcowa wersja ma się pojawić za kilka miesięcy.

Czy będziemy w stanie przetrawić tę kolejną porcję papki dla oczu?

Poprawka z ArtDepartmenta, czyli karty łączcie się.

Znowu pojawia się kolejna, bardziej udoskonalona wersja programu ArtDepartment Pro. Tych, którzy nie mają w swoich Amigach wielu megabajtów i twardego dysku informuję, że jest to program do przekształcania wszelakich grafik na inne, równie ładne, a może jeszcze ładniejsze. Nowa wersja nosi symbol 2.5 i jest najbardziej znamienitą poprawką w historii programu.

W programie pojawił się najnowszy krzyk mody, czyli GUI. Jest to graficzny panel porozumiewania z użytkownikiem (graphical user interface). Oprócz niego pozostawiono dla bardziej konserwatywnych użytkowników panel sterowany guzikami.

Program współdziała z commodorowskim standardem RTG (retargetable graphics), który ma zjednoczyć wszystkie karty graficzne. Dzięki niemu możliwe jest wyświetlanie grafik bezpośrednio przez takie karty jak: Picasso, EGS, Retina, czy VideoToaster. Dodano również nowe formaty animacji: CDXL, FLC i FLI oraz JStream i ICO dla bezpośredniego odczytu i zapisu Workbenchowych ikonek.

Wersja 2.5 ma możliwość wykorzystywania specjalnego trybu sublimacyjnego nowej drukarki kolorowej Fargo Primera w taki sposób, że wydruki wydają się rywalizować z fotografiami. Drukarka podobno kosztuje 1000$, tak więc gra jest warta świeczki.

Wszyscy zwolennicy ARexxa z pewnością nie zawiodą się na 100 nowych skryptach bazujących na tym programiku.

Dzięki nim możliwe jest wykorzystanie pełnych efektów "full motion".

Obszerna dokumentacja z wieloma przykładami, poradami, ilustracjami i rozszerzoną częścią poświęconą ARexxowi zamyka całość.

Cena programu wynosi 295$, zaś cena upgrade'u 45$.

Amiga 64

Wielu z Was przesiadło się na Amigę ze starego dobrego komodorka C64. Niektórzy ciągle tęsknią. Na Amigę powstało wiele emulacji tej 8-bitowej maszynki umożliwiających nawet odpalanie sporej ilości gier.

Ostatnio Questronix wypuścił na rynek komercyjną wersję emulatora w pakiecie o "dźwięcznej" nazwie A64. Pakiet zawiera przystaweczkę, która pozwoli Wam na podłączenie wszystkich peryferii od C64 oraz 104 stronicową instrukcję. Nowa wersja zapewnia pełną emulację starego systemu. Jedynie posiadacze Amig wyposażonych w AGA będą musieli na wstępie przełączyć tryb wyświetlania. Poza grafiką możliwa jest pełna emulacja dźwięku i korzystanie z fast-loadera. Dla Amig opartych na procesorze 68000 zaimplementowano również GEOSa. Żeby komodorkowi nie było smutno, umożliwiono mu funkcjonowanie z dodatkiem REU (RAM Expansion Unit), który podbija ilość pamięci ze 128kB do 2MB.

W pakiecie istnieje możliwość transferu danych z dysków 5 1/4 do Amigi, a także tłumaczenia programów z C64 na szybkie kody 68000. Cena pakietu nie jest nam znana.

Trójwymiarowe dyski CD

Piractwo kwitnie również wśród dysków CD na Amigi i nie tylko. Wprawdzie dysków tych nie można skopiować domowym sposobem (jeszcze?), ale robią to lepiej wyposażeni piraci. Szacuje się, że w zeszłym roku rozprowadzono nielegalnie 700.000 kompaktów. Oprócz tego, że są one robione na sprzęcie niższej jakości i częściej mogą się na nich zdarzać błędy, niczym nie różnią się od oryginałów. Gry komputerowe na CD nie są jeszcze masowo kopiowane, ale kto wie jak długo potrwa ta cisza przed burzą.

Proponuje się drukowanie na dyskach grafik holograficznych, które będą niezwykle trudne do podrobienia. Brak takiego hologramu nie wpłynie na działanie pirackiej kopii, ale pozwoli nabywcy zorientować się, że ma do czynienia z podróbką. Ciekawe, kiedy zostaną wprowadzone trójwymiarowe dyskietki?





ODWIECZNY problem z pamięcią

OUT OF MEMORY

Tomasz Blanc

Porównując komputery domowe z innymi na co dzień wykorzystywanymi przez nas urządzeniami, trafnym wydaje się spostrzeżenie, że znajdują się one w dalszym ciągu w fazie rozwoju pierwotnego. Jest to jednak rozwój niesamowicie szybki i błyskotliwy w stosunku do innych dziedzin. Nieco więcej niż trzynaście lat temu, komputer utożsamiany był najczęściej z kalkulatorem kieszonkowym wykonującym podstawowe działania matematyczne.

Przez ten krótki okres nastąpił olbrzymi rozwój możliwości tych maszynek. Doprowadziło to w efekcie do powstania komputerów domowych naszych czasów. Monochromatyczne wyświetlacze ciekłokrystaliczne zastąpione zostały monitorami kolorowymi o niemal fotograficznej dokładności odzwierciedlania obrazu. Prymitywne sygnały dźwiękowe, które zachwycały pierwszych fanów komputerowych, dziś przyprawiłyby ich raczej o ból głowy.

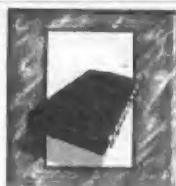
Mimo osiągnięcia tak wspaniałych efektów w dziedzinie technologii komputerowych, nie przewiduje się zahamowania rozwou. Wydaje się, że jesteśmy skazani na nieustanny rozwój. Coraz to nowsze modele komputerów wymagają coraz bardziej wyrafinowanego oprogramowania, wykorzystującego w pełni ich wzrastające mo żliwości. Jednym z elementów podlegających ciągłemu wzrostowi jest pojemność komputera - czyli jego pamięć.

Wracając do czasów, gdy na rynek wkroczyły pierwsze modele komputerów domowych (ZX81 i VIC20), 5KB pamięci traktowano jako ilość wprost ekstrawagancko dużą. Prawdziwym szokiem było zaprezentowanie pierwszego modelu C-64, którego ze względu na niewyobrażalną wówczas ilość pamięci (64KB) porównywano do słonia. Można sobie wyobrazić jakie wrażenie wywoływał wówczas model C-128. Jeszcze do dziś pamiętam z jakim szacunkiem spoglądało się na te wspaniałe maszyny, jakie legendy krążyły o ich niebywałych możliwościach (choć w zasadzie niewielu ludzi wiedziało jak je wykorzystać). Od tego czasu stało się niemal regułą, że każdy kolejny wprowadzany na rynek komputer posiadał dwa razy większą pojemność pamięci od swojego poprzednika.

Początki Amigi były pod tym względem podobne. Pierwsze modele A1000 wyposażone były w 256KB pamięci - co wystarczało jedynie na uruchomienie systemu wraz z jednym, niezbyt wymagającym programem. Wkrótce przekonano się, że jest to ilość śmiesznie mała. Kolejne modele produkowano już z pamięcią o pojemności 512KB. Taką pamięć posiadały też pierwsze modele A500 i przez jakiś czas sytuacja się ustabilizowała. Były to czasy, gdy Amigi używane były w domach głównie do gier. Wraz z postępującym napływem poważ nych programów użytkowych (a zwłasz cza graficznych), komputery te zaczęto chętnie wykorzystywać również do celów profesjonalnych (głównie w dziedzinie video). Tu jednak ponownie pojawił się problem brakującej pamięci. Powstało szereg rozszerzeń pozwala jących zwiększyć pojemność komputera do 1MB, a firma Commodore uznała tę wielkość za podstawową i rozpoczęła produkcję nowych modeli (A500+, A600) wyposażonych właśnie w taką ilość pamięci. Wraz z zastosowaniem w A1200, szybszego procesora i rozbudowanych układów graficznych, zwiększyła się również do 2MB ilość podstawowej pamięci. Jeszcze kilka lat temu była to wartość niewyobrażalna, a umieszczenie jej we wnętrzu komputera jako standard odczytano by jako stanowczo nadmierną rozrzutność.

Czasy się zmieniają i obecnie, nawet 2MB pamięci nie wystarczają, jeżeli pragniemy wykorzystywać swoją Amigę w sposób poważny. Oczywiście wielkość ta wystarczy do uruchomienia większości gier i kilku poważnych aplikacji, ale wydaje się śmieszna podczas zabawy z grafiką. Nowe tryby graficzne, umożliwiające uzyskanie wspaniałych efektów, bezlitośnie pochłaniają setki kilobajtów pamięci. Wyświetlenie obrazu w trybie HAM-8 o wysokiej rozdzielczości (1280x512) zajmuje około 650KB! Jeżeli dodamy do tego pamięć niezbędną do uruchomienia i działania WorkBencha - stracimy już cały 1MB pamięci, zanim jeszcze zaczniemy cokolwiek robić.

Mówiąc oględnie, pamięć Amigi podzielona została na dwie przestrzenie: FAST i CHIP. Pamięć FAST przeznaczona jest do wyłącznego użytku procesora głównego. Ponieważ inne układy nie mają dostępu do tego obszaru, procesor może pracować na tym obszarze z pełną szybkością (dlatego pamięć ta nazywa się 'Fast' - szybka). Jest to miejsce składowania programów. Drugim obszarem pamięci Amigi jest CHIP - odgrywający dosyć specyficzną rolę - jest to jedyny obszar pamięci, do którego dostęp mają specjalistyczne układy Amigi. Bez pamięci CHIP, Paula, Agnus i Denise (oraz ich



nowe odpowiedniki) nie mogłyby zdać świadectwa swojej działalności - wspaniały dźwięk i grafika Amigi, nigdy nie dotarłyby do naszych receptorów i nikt nie zobaczyłby wspaniałych efektów funkcjonowania komputera.

Faktycznie każdy program uruchamiany na Amidze robi użytek z pamięci CHIP - jeżeli, na przykład uruchamiasz grę, cała grafika i dźwięk, które przeważnie są jej podstawą, składowane są właśnie w pamięci CHIP. Jednak nie tylko programy 'rozrywkowe' wymagają obecności tej pamięci. Jeżeli wykorzystujesz bardziej poważne programy (bazy danych, edytory tekstu, arkusze kalkulacyjne), CHIP jest automatycznie używany do składowania ekranów programu, dzięki czemu możliwy jest do nich szybki dostęp. Jedynymi programami nie wymagającymi tego rodzaju pamięci, są programy uruchamiane i działające z poziomu CLI. Ale nawet dane okna Shell, z którego zostają one przeważnie uruchamiane, muszą być umieszczone gdzieś w pamięci CHIP, aby umożliwić Denis'owi narysowanie właściwego obrazu ekranu.

Pierwsze modele Amigi były w stanie zarządzać maksymalnie 512KB pamięci CHIP. Była to ilość wystarczająca do większości gier i mniej wymagających programów, lecz obecnie ilość 1MB lub nawet 2MB CHIP traktowana jest jako standard. Nowa A1200, na przykład, zawiera właśnie 2MB pamięci CHIP. A600 i A500+ może zostać również rozbudowana do tej wielkości nie wystarczającej jednak do pracy z większością zaawansowanych programów graficznych (Imagine) oraz profesjonalnych programów DTP (ProPage).

O ile w przypadku większości "zwykłych" komputerów problem braku pamięci nie jest od razu zauważalny, w przypadku Amigi wydaje się on szczególnie dokuczliwy. Związane jest to z możliwością pracy tego komputera w trybie wielozadaniowym (multitaskingu). Multitasking to wspaniała rzecz, ale jest również najszybszą drogą zapełniania pamięci. Uruchomienie wielu programów jednocześnie na jednym komputerze stwarza wiele nowych możliwości, przysparzając jednak problemu braku pamięci. Uruchamiając kilka aplikacji, można z wykorzystaniem Arexxa, zapewnić pomiędzy nimi komunikację sprowadzającą się do wymiany danych. Przyjemność takiej pracy pozostaje jednak w sferze marzeń dla wielu posiadaczy Amigi. Ograniczeniem jest tu ilość posiadanej pamięci, która w tym przypadku powinna wynosić więcej niż 4MB. Choć nieco przewrotne, prawdziwe jest stwierdzenie, że w przypadku pracy wielozadaniowej, pamięci nigdy za wiele i zawsze jej brakuje.

Oczywiście w każdej chwili można zdecydować się na rozszerzenie pamięci komputera. Zanim jednak do tego dojdzie, należy zastanowić się nad paroma ważnymi sprawami. Pierwsze pytanie, jakie powinniśmy sobie postawić to: ile pamięci potrzeba? Należy przy tym pamiętać o jednej żelaznej zasadzie: im więcej pamięci posiadam, tym więcej mogę zrobić (i tym więcej chcę zrobić). Wymagania będą stale wzrastać, i nawet jeżeli rozszerzysz swój komputer o 4MB, może nadejść taki moment, gdy będziesz potrzebował więcej. Ideałem byłoby więc, gdyby można zakupić taką ilość pamięci, która przewyższa nasze aktualne potrzeby. To jednak jest często niemożliwe, gdyż zwykle potrzeby przewyższają nasze możliwości. Podstawowym ograniczeniem jest oczywiście koszt takiego rozszerzenia. Aby pomóc Ci w dokonaniu wyboru, krótko opiszę możliwości Amigi wyposażonej w różne konfiguracje pamięci. Zacznę od najmniejszej - 1MB.

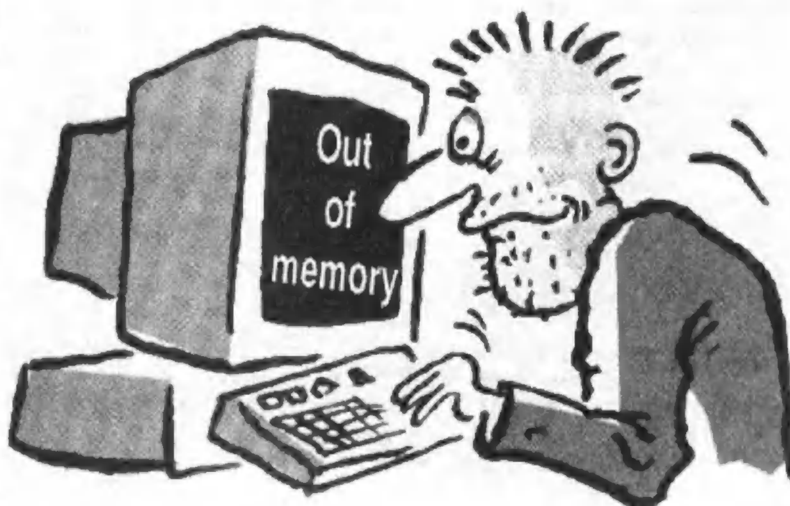
1MB: Przy tej ilości pamięci praktycznie wszystkie gry uruchamiają się poprawnie. Również większość programów użytkowych może w niewielkim stopniu zostać wykorzystana. Większość z nich nie pracuje jednak "pełną parą". Np. podczas pracy z DPaintem niemożliwe jest otworzenie trybu grafiki wysokiej rozdzielczości przy większej ilości kolorów, znacznie ograniczone są również możliwości tworzenia animacji. Ogólnie rzecz biorąc, brakuje pamięci dla danych wszelkiego rodzaju. W przypadku programów graficznych, utrudnione będzie ładowanie skomplikowanych rysunków, w przypadku programów muzycznych - długich i pamięciożernych próbek dźwięku. Niektóre programy, takie jak Art Department Pro czy Pro Page 4 nie zostaną nawet uruchomione.

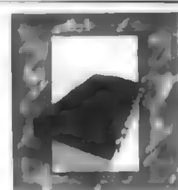
2MB: Programy takie jak DPaint 4 (w wersji 'AGA') będą uruchamiać się poprawnie, ale możliwości kreowania animacji będą nadal ograniczone. Jednak w tym przypadku większość aplikacji uruchamia się poprawnie. Do profesjonalnego wykorzystania tych największych brakuje jednak jeszcze kilku MB pamięci.

4MB: W tym przypadku, większość dużych programów uruchamia się i funkcjonuje w sposób satysfakcjonujący. Jednak próby jednoczesnego uruchomienia dwóch takich programów kończą się najczęściej niepowodzeniem. Od tej ilości pamięci rozpoczyna się również możliwość pracy na grafice 24-bitowej.

6MB: Wielu profesjonalnych użytkowników Amigi uważa tą ilość za absolutne minimum do wykorzystania w pełni możliwości tego komputera. W tym przypadku praca w systemie wielozadaniowym nie jest już właściwie ograniczona. Można swobodnie uruchamiać i przesyłać dane pomiędzy kilkoma uruchomionymi programami.

9MB: Jest to wielkość, do jakiej maksymalnie można rozszerzyć większość dotychczas produkowanych modeli Amigi. Tu właściwie można robić wszystko, nie martwiąc się nawet o składowanie danych. ❑





AMIQUEST 3270

Tomasz Kulbacki

Każdy twardy dysk staje się za mały. Jeśli nie natychmiast po kupnie, to niedługo potem. Nie jest to kolejne prawo Murphy'ego, lecz ponura rzeczywistość. Ci z Was, którzy używają Amigi z twardzielem z pewnością będą musieli kiedyś dokonać trudnego wyboru: kupić drugi twardy dysk, bądź też sprzedać stary i kupić większy. Właściciele A600 i A1200 byli dotychczas skazani na drugie rozwiązanie, gdyż komputery te nie posiadają miejsca na umieszczenie drugiego napędu wewnątrz komputera. Ze względu na (oględnie mówiąc) dość nieprzyjemne ceny dysków 2,5" jest to rozwiązanie zmuszające do baaardzo głębokiego sięgnięcia do kieszeni. Alternatywą jest instalacja dysku 3,5", która też nie zawsze odbywa się bez kłopotów (słaby zasilacz). Pewną nadzieję na zmianę tego stanu rzeczy niesie oferta francuskiej firmy Archos, która wypuściła w październiku ubr. zewnętrzny kontroler AT-BUS podłączany do złącza PCMCIA.

AmiQuest to kontroler IDE (AT-Bus) umieszczony w czarnym pudełku o kształcie dopasowanym do obudowy A600/1200, podłączany do komputera za pośrednictwem złącza PCMCIA. W tym pudełku [redacted] wymiarach 4 x 10 cm można ulokować 2,5" dysk twardy. Do kontrolera dołączone jest oprogramowanie o funkcjach właściwie identycznych z commodorowskim HD-Toolbox. AQToolbox służy do ręcznego ustawiania parametrów pracy (podziału na partycje, ustawienia priorytetów, Autoboot, Automount, itd.). AQSetup wykonuje te same funkcje całkowicie automatycznie, natomiast AQInstall instaluje oprogramowanie kontrolera na dysk.

Przy wyborze twardego dysku obowiązują takie same ograniczenia, jak przy zakupie dysku do kontrolera wewnętrznego A600/A1200. Oznacza to, że poprawnie będą działać tylko dyski Seagate, Conner i Western Digital (nie wszystkie). Osiągi AmiQuesta są porównywalne z wewnętrznym kontrolerem commodorowskim. Różnice rzędu 1% na jego niekorzyść są praktycznie niezauważalne.

AmiQuest w swojej obecnej postaci nie jest rewelacją. Nie rozwiązuje bowiem głównego problemu posiadacza A600/1200, który zazwyczaj marzy o bezproblemowym użytkowaniu taniego dysku 3,5". Jest za to kapitalnym ułatwieniem [redacted] przenoszeniu danych między komputerami.

Archos zapowiada wkrótce wypuszczenie analogicznego kontrolera przystosowanego do instalacji dysku 3,5" i wyposażonego we własny zasilacz. Jeśli te zapowiedzi sprawdzą się, dopiero ta wersja AmiQuesta będzie miała duże szanse odniesienia sukcesu rynkowego.

AmiQuest 2,5" kosztuje w RFN ok. 350 DM (bez twardego dysku). ❑



Supra TURBO 28

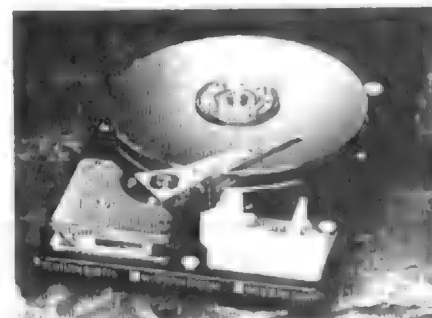
Tomasz Kulbacki

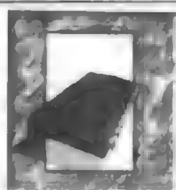
Można pomyśleć, że w epoce Amig 4000, 1200 i superszybkich kart turbo z procesorami 68030 lub 68040 przyspieszacze oparte na beznadziejnie przestarzałym procesorze MC 68000 znikną z rynku w szybkim tempie. Żaden [redacted] znajdujących się w sprzedaży modeli nie wywiera specjalnego wrażenia swoimi osiągami. Z tego powodu należałoby sądzić, że nie ma firmy, która chciałaby podobny produkt wypuścić. Tym większe było moje zdziwienie, gdy zobaczyłem reklamówkę firmy Supra zachwalającą kartę Supra Turbo 28. Z początku sądziłem, że jest to kolejna pseudoturbokarta dla ubogich, ale po dokładnych oględzinach szybko zmieniłem zdanie.

Supra Turbo [redacted] jest oferowana w dwóch wersjach: zewnętrznej - do Amigi 500(+) i wewnętrznej - jako karta Zorro II do Amigi 2000. Poza tą różnicą obie wersje są identyczne. Konstrukcja karty oparta jest [redacted] procesorze MC68HC000 taktowanym zegarem [redacted] częstotliwości 28 MHz z zewnętrznym buforem pamięci (Cache Memory). Karta nie ma pamięci Flash do kopiowania Kickstartu, ani własnej pamięci RAM. Technika wykonania jest tak samo prosta, oszczędna, [redacted] zarazem genialna. Test Benchmark wykonany programem AIBB 6.1 wykazuje, że Karta Supra Turbo 28 w teście Graphics osiąga wyniki 2,16 razy lepsze od zwykłej pięćsetki z 1 MB CHIP i 4 MB FAST RAM. Ten wynik pozostawia daleko w tyle nawet karty turbo do A500 oparte na procesorze MC68020 i 68030. W testach Integer i FP wyniki [redacted] przyczyn obiektywnych nie są już tak szokujące, ale zawsze dorównują osiągom uzyskanym na kartach z MC 68020 bez koprocesora.

Supra Turbo [redacted] charakteryzuje się bardzo dobrą kompatybilnością zarówno pod względem programowym jak i sprzętowym (brak konfliktów z innymi rozszerzeniami sprzętowymi). Dołączona instrukcja informuje jedynie, że przy pracy z rozszerzeniami z DMA-Masterem należy odłączyć (specjalnie do tego przeznaczonym jumperem) buforowanie FAST-RAMu. Oprócz tego, karta nie pracuje z kontrolerami SCSI II firmy GVP wyposażonymi w ROMy wersji niższej niż 3.12. W celu zapewnienia poprawnej pracy z Amigą 500(+), Supra Turbo powinna być umieszczona jako pierwsze urządzenie podłączone do slotu Expansion. Gdyby trafił się jakiś program odmawiający współpracy, kartę można odłączyć wyłącznikiem.

Supra Turbo [redacted] jest wykonana w technice SMD. Dołączana jest do niej instrukcja obsługi w języku niemieckim lub angielskim i nieco oprogramowania. Cena karty wynosi [redacted] RFN ok. 300 DM. ❑





KARTY PCMCIA

Tomasz Kulbacki

Przed rokiem wraz z wejściem na rynek komputerów Amiga 600, użytkownicy Amigi mieli okazję poznać nowy rodzaj nośnika pamięci dostępny dotychczas tylko dla użytkowników pecetów w wersji Laptop i Notebook - kartami PCMCIA.

Standard PCMCIA (Personal Computer Memory Card Interface Adaptor) został stworzony z myślą o zainstalowaniu w przenośnych komputerach klasy PC zasilanych bateryjnie. Działo się to w czasach, kiedy twarde dyski były jeszcze stosunkowo duże (tzn. duże były ich wymiary zewnętrzne, a nie pojemność) i bardzo prądożerne. Karty PCMCIA jako nośnik bardzo energooszczędny i oferujący korzystny jak na tamte czasy stosunek pojemności pamięci do wymiarów oraz ilości zużywanej energii, okazały się rozwiązaniem na tyle atrakcyjnym, aby skłonić niektórych producentów do ustalenia standardu złącza i jego seryjnego montażu w komputerach. Początkowo obowiązywał standard 68-pinowego złącza, co całkowicie wystarczało w erze dominacji procesora Intel 80286 o 16-bitowej szynie danych i szynie adresowej. Jednak okazało się to niewystarczające po upowszechnieniu się procesorów 32-bitowych, gdyż 68 pinów nie pozwalało na zastosowanie w kartach PCMCIA szybkich pamięci 32-bitowych.

Karty PCMCIA, które działały poprawnie w komputerach 16-bitowych (np. PC 80286 albo Amiga 600), okazały się beznadziejnie przestarzałe po wejściu na rynek komputerów opartych na procesorze 32-bitowym. Kolejną wadą tego standardu, spowodowaną zbyt małą liczbą pinów, był brak możliwości montażu typowych pamięci dynamicznych DRAM, co zmusiło producentów do używania o wiele droższych pamięci statycznych. Problem ten został rozwiązany wraz z wejściem na rynek pamięci pseudostatycznych PSRAM. Upowszechnienie komputerów 32-bitowych wymusiło więc ewolucję standardu PCMCIA i tak powstał typ PCMCIA II o 32-bitowej szynie danych pozwalający na nieograniczone korzystanie z pamięci dynamicznych.

Ci z Was, którzy w tym miejscu niecierpliwie oczekują na informację, który typ złącza zastosowano w Amidze 600 i 1200, przeżyją teraz chwilę bolesnego rozczarowania. Commodore w pogoni za obniżką kosztów zastosował oczywiście stary typ interfejsu, co automatycznie dyskwalifikuje karty PCMCIA jako rozszerzenie RAM w A1200. Procesor 32-bitowy pracujący na 16-bitowej pamięci zachowuje się mniej więcej tak, jak procesor 16-bitowy, a mówiąc wprost - działa beznadziejnie wolno. Dlatego posiadaczom A1200 radzę zaopatrzyć się w rozszerzenie montowane "pod klapką". Commodore podaje wprawdzie, że można tam zainstalować maksymalnie 4 MB RAM, ale pomysłowi konstruktorzy rozszerzeń sprzętowych już teraz oferują karty 0/2/4/8 MB z możliwością zamontowania koprocesora MC 68881/2. Inaczej sytuacja wygląda w przypadku modelu A600. Karty PCMCIA dają możliwość instalacji pamięci FAST, a 16-bitowa architektura procesora MC68000 jest optymalna do pracy z wszystkimi rodzajami kart.

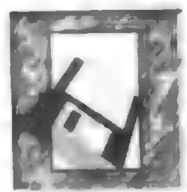
Krótki przegląd dostępnych kart PCMCIA.

1. Karty PSRAM (Pseudo Static Random Acces Memory). Po zamontowaniu pełnią funkcję 16-bitowego rozszerzenia pamięci RAM o pojemności 2 lub 4 MB. Konfigurują się w pełni automatycznie po umieszczeniu w porcie i uruchomieniu komputera.

2. Karty SRAM (Static Random Access Memory). Pomimo brzydkiej nazwy, są to najbardziej uniwersalne i najdroższe z kart dostępnych na rynku. Mają dwojakie zastosowanie. Mogą funkcjonować jako normalne rozszerzenie FAST-RAM (podobnie jak PSRAM), bądź też jako podtrzymywany bateryjnie RAM-dysk. Karta PSRAM używana jako RAM-dysk nie różni się sposobem odczytu i zapisu informacji od zwykłej dyskietki, ani twardego dysku. Przed pierwszym użyciem musi być "sformatowana" specjalnym programem ("PrepCard" z katalogu "Tools") i podobnie jak dyskietka jest podzielona na cylindry. Nie brakuje nawet takiego drobiazgu jak zabezpieczenie zawartości przed kasowaniem i zapisem. Zamontowana na karcie bateria zapewnia podtrzymanie zawartości pamięci nawet po wyjęciu z komputera. Okres jej trwałości wynosi 2-3 lata.

3. Karty ROM (Read Only Memory). Karty ROM działają analogicznie do starego dobrego Cartridge znanego z Commodore 64 i Atari XE. Po włożeniu karty do slotu i włączeniu komputera, program zapisany w ROM zostaje skopiowany do pamięci operacyjnej komputera i uruchomiony.

Na niemieckim rynku potentatem w dziedzinie kart PCMCIA jest firma Boeder, dotychczas znana jako producent papieru do drukarek i pudełek na dyskietki. Oferuje ona wszystkie rodzaje kart PCMCIA i zapowiada wypuszczenie na rynek w najbliższej przyszłości oprogramowania umożliwiającego uruchamianie programów zapisanych na kartach ROM, bezpośrednio na nich, bez korzystania z pamięci operacyjnej komputera. ❑



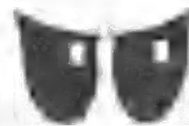
SUPER MEMO SUPER

Tomasz Kokoszczyński

Znana i legendarna firma Twin Spark Soft [redacted] z redakcją AMIGOWCA ogłosiła w zeszłym numerze naszego pisma wielki konkurs na najlepszą bazę danych dla programu SuperMemo. Ci, którzy [redacted]ją program wiedzą co jest [redacted]. Ci, którzy go nie poznali, [redacted] tylko gdzieś o uszy obiły się jakieś pogłoski, traktują go prawie jak legendę. Bo faktycznie jest to już legenda. Program choć młody zaliczył wiele prestiżowych konkursów i dotarł [redacted] nich do finału (SOFTWARE FOR EUREOPE CeBIT 92, SOFTARG 92). Teraz zaś jego twórcy zaczęli przyznawać nagrody...

Program SuperMemo opiera się [redacted] naukowo sprawdzonej metodzie jak najbardziej efektywnego nauczania. Nie oznacza to oczywiście, że program nauczy się za nas, pójdzie do szkoły i napisze klasówkę na 5. Nic z tych rzeczy. Jest to jednak doskonałe narzędzie, któ[redacted] pozwoli nam bez problemów usystematyzować naszą pracę. Oprócz tego program jest doskonałym źródłem wszelkiej informacji od słówek w języku niemieckim, czy angielskim poczynając, na znakach drogowych oraz obrazach wielkich mistrzów kończąc. Ładne porównanie, nie ma co, ale tak to już jest w SuperMemo - można się z niego dowiedzieć o wszystkim (patrz tabelka).

Jak stąd widać zgromadzono całą furę informacji i teraz należy się jej dzielnie wkuć, co zapewnia nam rozrywki na kilka lat do przodu. Tak, tak. Nikt Wam nie gwarantuje, że staniecie się nagle geniuszami przy korzystaniu z programu SuperMemo. Autorzy zastrzegają jedynie, że program pozwoli Wam [redacted] wykorzystanie możliwości Waszego mózgu w sposób optymalny. I tyle. No i zapewniają, że zestaw baz programu będzie poszerzany o nowe, nowsze i najnowsze. Oczywiście tylko pod warunkiem, że zafundujemy sobie legalną kopię programu i legalne kopie baz, bo do interesu nikt nie będzie dopłacał, a jak się firma wycofa to się już nic nie poszerzy. Poza tym musicie wiedzieć, że wersje amigowskie programu, bo przecież przedtem istniał on jedynie na IBMa, są o wiele tańsze. I w tym momencie opłaca się mieć Amigę...



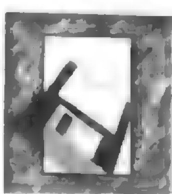
Program dostarczany jest w estetycznym, przezroczystym opakowaniu wraz z instrukcją, kartą rejestracyjną i dyskietką instalacyjną. Wszystko oczywiście jest kontrolowane przez wszechobecne oczy - znak firmowy Twin Spark Softu.

Ewentualne zainstalowanie programu na twardym dysku nie powinno sprawić większych problemów - firma dołączyła specjalny instaler. Musimy tylko w odpowiednim miejscu umieścić szufladkę DB. Musi się ona znajdować w urządzeniu logicznym SuperMemo:. Ot i cała instalacja. Gdyby jeszcze wyniknęły jakieś problemy od czego mamy instrukcję. Program został sprytnie zabezpieczony. Nie jest to zabezpieczenie z rodzaju uciążliwego wyszukiwania stron z instrukcji i wpisywania 3 wyrazu w siódmym wierszu...

Instrukcja wprowadza nas w zasadę działania SuperMemo. Jest to pro

Tabelka: Bazy danych dostępne na [redacted] 01.02.1994

L.P.	Baza danych	L.P.	Baza danych
1.	Astrologia	14.	Czasowniki mocne i nieregularne
2.	Historia	15.	w języku niemieckim
3.	Poczet królów polskich AGA	16.	Język hiszpański
4.	Podstawy języka angielskiego	17.	Ortografia języka polskiego
5.	Advanced English	18.	Angielski w technice komputerowej
6.	Zaawansowane słownictwo	19.	First Certificate in English Medical
7.	Angielski w biznesie	20.	English Medical Biochemistry and Biology
[redacted]	Angielski dla nerwusów	21.	Angielski w motoryzacji
9.	Technical English	22.	Język francuski
10.	Angielskie idiomy	23.	Biologia dla kandydatów [redacted] AM
11.	English Grammar	24.	Choroby wewnętrzne
12.	Esperanto	25.	Capitals of the World
13.	Język niemiecki		



Data
Dzień
Zapamiętane
Nietknięte
Razem
Zaległe
Obciążen.+/-
Obciążenie
Średni czas
Nakład pracy
Odstęp +/-
Odstęp
Współczynnik
Powtórka
Upływ +/-
Upływ ()
Retencja
Ocena
Czas

gram służący do uczenia się dowolnej partii materiału i aby jego zapamiętywanie było lepsze, musimy kierować się dwoma metodami:

* Odstępy między powtórzeniami powinny być na tyle długie, żebyśmy się nie zamordowali ciągłym powtarzaniem tego samego "w kółko Macieju".
* Odstępy muszą być [redacted] tyle krótkie, żeby gromadzona wiedza nie wyleciała nam bezpowrotnie [redacted] głowy.

Prosto powiedzieć, trochę trudniej zrobić. Można sobie to oczywiście spróbować ustalić eksperymentalnie, ale taki eksperyment może trwać całe życie i nic [redacted] tego nie wyjdzie. Lepiej wykorzystać możliwości SuperMemo.

Program działa na bazie specjalnych przeliczników. Sprawdza ilość zapamiętywanego przez Ciebie materiału i ustala częstość, zakres i liczbę powtórek. Wszystko oczywiście dla Twojego dobra. Informacje są podawane z boku ekranu, ale myślę, że nie jest to najbardziej interesująca część programu.

Bardziej interesujące jest to, jak możemy się uczyć. Na początku należy otworzyć oczywiście jakąś bazę z danymi. Radzę zacząć od tej, która nam się najbardziej podoba. Jednorazowo oczywiście możemy pracować tylko z jedną bazą. Po jej otwarciu możemy z menu Jednostka wybrać opcję Nauka. No i zaczynamy się uczyć...

Program pokazuje nam słowo, grafikę, czy hasło - my zastanawiamy się przez chwilę z czym się nam to kojarzy i naciskamy guzik Odpw. Pojawia się odpowiedź i teraz musimy siebie ocenić. Na początku jest to najtrudniejsze zadanie wybrać [redacted] 6 stopniowej skali [redacted] (0-6) to co odpowiada stanowi naszej wiedzy na zadane hasło. Oczywiście nie ma problemu jeżeli odpowiedź znamy na piątkę. Gorzej, gdy coś nam świta, ale nie wiemy "w którym kościele dzwoni". Wtedy trzeba wybierać pomiędzy 3-2-1, itp. No zera też się czasami zdarzają, ale są rzadkie i chyba nie musimy być tacy surowi dla siebie?

Program to wszystko notuje, sumuje i ustala termin następnej powtórki. Naprawdę trudno jest go przechytrzyć. Następna powtórka może wypaść oczywiście nazajutrz, ale też może się zdarzyć, że będziemy mieli kilka dni spokoju. Zawartość baz jest praktycznie nieograniczona. Oprócz wszelakiego rodzaju słówek i haseł z różnych dziedzin wiedzy (jeszcze raz przypomnij sobie tabelkę), mamy tu możliwość wstawiania różnego rodzaju grafik, czy nawet obrazów wielkich mistrzów.

Jest to dosyć pomocne np. przy uczeniu się znaków drogowych i podstaw ruchu. No bo przecież nawet najlepszy opis nie odda tego, co można przedstawić w krótkiej grafice.

Jednak głównym zadaniem programu jest nauczenie nas słownictwa, idiomów i niektórych pojęć. Hasła z zakresu wielu nauk są troszeczkę na wyrost - no bo przecież wiedza nie polega na zebraniu w głowie masy definicji i ich wypowiadaniu przy stosownych okazjach. To nie jest mądrość! Oczywiście zadanie podstawowe program spełnia wspaniale. Można w miarę krótkim czasie opanować podstawowy zakres słownictwa danego języka. Oczywiście program nie zastąpi nas w jego używaniu - słownictwa tego przecież nie będziemy powtarzać do końca życia przy pomocy programu SuperMemo. Nie uczymy się tu wiedzy dla wiedzy! SuperMemo buduje nam szkielet wiedzy w danym przedmiocie, pomaga poznać słownictwo, które ułatwi nam lekturę odpowiednich pozycji w danym zakresie.

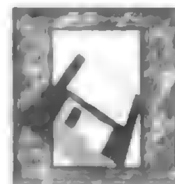
W programie istnieje również ciekawa możliwość dopisywania własnych haseł, czy słówek. Można też komponować bazy danych składające się z własnych danych, ponieważ grafiki nie są zapisywane w jakimś specjalnym formacie tylko w zwykłym IFFie. Troszkę można by tu ponarzekać na edytor, który czasami wyrzuca nam kawałek linii poza ekran i trudno jest się do niego dostać, ale to się da przeżyć. Poza tym opcje wstawiania pozycji, kopiowania i przepisywania są ułożone starannie i obsługiwane przy pomocy pojedynczych klawiszy.

Przy ciągłej pracy z bazami danych musimy pamiętać o ich bezpieczeństwie. Nie zapomnieli o tym też autorzy wyposażając program w opcje zabezpieczające. Przy błędnym zapisaniu pliku (np. gdy wywołamy guru w trakcie działania programu), może pojawić się informacja, że baza została nieprawidłowo zamknięta. Ale nic to - odpowiednia opcja zerowania bazy załatwi

SuperMemo v1.12A by TWIN SPARK SOFT. Copyright © 1993 SuperMemo World.

Data
Dzień
Zapamiętane
Nietknięte
Razem
Zaległe
Obciążen.+/-
Obciążenie
Średni czas
Nakład pracy
Odstęp +/-
Odstęp
Współczynnik
Powtórka
Upływ +/-
Upływ ()
Retencja
Ocena
Czas

SuperMemo
UCZ SIĘ SZYBKO
I ZAPOMNIJ O ZAPOMINANIU
SuperMemo
World
Wersja na Amigę opracowana przez
TWIN SPARK SOFT.



sprawę (oczywiście wszystkie nasze naukowe grzechy zostaną w ten sposób odpuszczone i będziemy usprawiedliwieni przed rodzicami).

Program [illegible] jedną zasadniczą wadę: motywacja. Nie możena jej zastąpić w żaden sposób. To my sami musimy się zmotywować i nie ma na to innej rady. Nie pomoże żaden najlepszy program. Dla tych, którzy mają skłonności do depresji może to być dodatkowy kamyczek do ich ogródka - tu tyle ciekawych rzeczy do nauki, [illegible] ja nie mam na nic ochoty. Wtedy drodzy rodzice zafundujcie dziecku więcej uwagi, [illegible] SuperMemo można będzie kupić jak wszystko wróci do normy.

Pozostałym dzieciom i ich rodzicom polecam program gorąco. Jest to coś, co powinno się znaleźć w każdym domu, w którym jest Amiga. Ten program wprawdzie nie uczy się za nas, ale uczy nas jak gospodarować swoim cza[illegible] i jak efektywnie uczyć się tego co wiedzieć trzeba. Program z pewnością przyda się wszystkim, którzy szukają pomocy do klasówek i egzaminów. Tylko [illegible] nic się zda SuperMemo na tydzień przed egzaminem. Tego po prostu nie da się tak szybko wchłonąć! ❑



AMI-MARKET

Zamienię rower górski+dopłata [illegible] Amigę [illegible] z modulatorem TV. **Rafał Romanowski, ul. Sos[illegible] 11a, 56-200 Góra (woj. leszczyńskie), tel. 43-34-40.**

Sprzedam oryginalne rozszerzenie pamięci 0.5 [illegible] produkcji Commodore (A501). Cena: 0.5 mln zł. Oferty listowne. **Arkadiusz Simkowski, os. Dolnośląskie 202/30, 97-400 Bełchatów.**

Sprzedam 2 [illegible] Fast Ram "Golden Image" do Amigi. Cena: 2 mln zł. lub zamienię na 2*1 MB SIMM. **Stanisław Ostrowski, ul. Zakątne 11, 59-230.**

Sprzedam Spectrum ZX+ [illegible] magnetofon. Cena: 700 tys. zł. **Marcin Krzewiński, ul. Stalowa 35, 03-425, tel. 19-38-59.**

Sprzedam programy na IBM i Amigę (5000 [illegible] 1 dysk). Dokładne informacje po przesłaniu koperty zwrotnej ze znaczkiem. **Szymon Sadowski, ul. Zielona 18, 62-040 Puszczykowo.**

Sprzedam Amigę [illegible] plus 2 MB Chip RAM, modulator, pokrywę, literaturę, oprogramowanie. Cena do negocjacji od 5.6 mln zł. **Tomasz [illegible]nasik, ul. [illegible] 1/28, 32-514 Jaworzno, tel. (0-35) 63-642.**

Kupię kolorowy monitor Philips do Amigi. Wymiana gier [illegible] Amigę 500. Kupię oryginał "Ishar I". **Marek Weczerek, ul. Kamionki 53, 44-295 Lyski (woj. katowickie).**

Sprzedam programy na Amigę (również w sys. VBS). Cena: 5000 zł. Katalog gratis. Amiga 1200-[illegible] 9.2 mln zł. **Arek Klimek, ul. Jana [illegible] II 18/3, 14-100 Ostróda, tel. 23-52.**

Amiga 500+2 MB Chip RAM, twardy dysk "Protar" 20 MB (SCSI), 0MB Fast, joystick, nowe użytki [illegible] twardym, 30 dysków. Okazja! Cena [illegible] mln zł. **[illegible] Kujawa, ul. Główna 1, 89-511 Cekcyn, tel. (Tuchola) 7114 (po godz. 20).**

Kupię drukarkę do Amigi 500, koniecznie kolorowa. Oferty z ceną i wydrukiem proszę kierować na adres: **Maciej Zawarski, ul. Kalinkowa 32e/69, [illegible] Grudziądz.**

Sprzedam drukarkę LC-10C, kompatybilna [illegible] C-64+programy. Mało używana. Cena 1.8 mln zł. **Piotr Dacko, ul. Widok 7/6, 70-773 Szczecin, tel. 613-030.**

Sprzedam Amiga Action Replay Mk3, literatura-990 tys. zł. **Piotr Laszczyk, Szeligówka 976, 34-511 Kościelisko, tel. 70-444.**

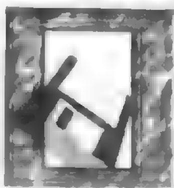
Sprzedam Amigę 500, 1 MB Fast-Chip, zegar, modulator za 4.35 mln zł. 170 dysków 5.25" DD Maxell. SKC D.Boxy-1.4 mln zł. **Krzysztof Nadowski, ul. Niepodległości 4/4, 38-300 Gorlice, tel. 221-90.**

Sprzedam Amigę 500, 2.5 MB, pokrywa, literatura, dyskietki, VII 93, modulator A520, stacja dysków Datalux SV702. Cena: 7.5 mln zł. **Wiesław Gawron, ul. Krakowska 1/37, 44-335 Jastrzębie, tel.(036) 719-911.**

Sprzedam A1200 na gwarancji + [illegible] joysticki, mysz, 40 dyskietek + pudełko, programy użytkowe. Cena: 9.5 mln zł. Stan bardzo dobry. **Bartosz Przymusiński, ul. Kanclerska 3/2, 60-327 Poznań, tel. 67-40-25.**

Sprzedam drukarkę Epson LQ-100 (24 igły) [illegible] traktorem i zapasową taśmą-10 miesięcy gwarancji. Cena: 5.6 mln zł. **Adam Zaborski, ul. Chopina 2/5, 05-800 Pruszków, tel. 58-65-03 (w godz. 19-22.30).**

Sprzedam Amigę 500, [illegible] MB, 1.3/2.0 Kick ROM, Chip-Fast, Bootselector, modulator, ok. 100 gier i programów użytkowych. Cena: 7 mln zł. **[illegible]towski Zbigniew, ul. Rolna 10, 78-200 Białogard, tel. 26-99.**



TURBO CALC

Tomasz [illegible]

Firma Stefan Ossowski's Schatztrue (Skrzynia Skarbów Stefana Ossowskiego) jest od wielu lat znana na rynku niemieckim. Zakres produktów firmy sięga od gry w szachy, skata, czy Monopol, poprzez programy typu Bundesliga i Lotto do programów profesjonalnych jak Formular Profi, Amiga Steuer, Euro-Übersetzer oraz omawiany dzisiaj Turbo Calc. Za każdym z tych programów stoi doświadczenie firmy i nazwisko jej założyciela. Ich reklamy zajmują po kilka stron najbardziej poczytnych niemieckich magazynów.

TurboCalc v.2.0 jest traktowany przez firmę na specjalnych zasadach. Jego promocja kosztuje sporo i jest przeprowadzana w oddzielnych reklamach, ale chyba warto, gdyż TC jest naprawdę dopracowany.

TurboCalc to nowoczesny arkusz kalkulacyjny umożliwiający nie tylko przeliczenia dolarów na złotówki na koncie naszego kooperanta pana N., ale pozwala też na tworzenie grafik do różnego rodzaju prezentacji na kongresach, zjazdach i spotkaniach roboczych. Jest to program, którego można używać zarówno w domu jak i w pracy. Nawet jeśli w pracy mamy IBMa wyposażonego w Excela, to jego dane mogą być przeniesione do TurboCalca i vice versa.

Oprócz obliczeń program może służyć jako zwykła, podręczna baza danych, w której można wpisywać adresy, lub inne, dowolne dane.

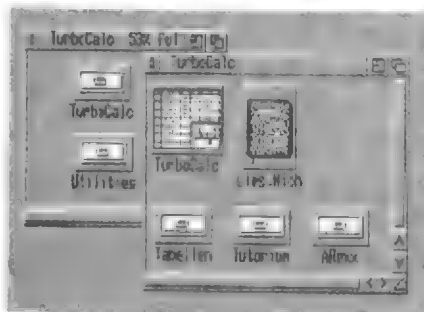
Program dostarczany jest w estetycznym pudełku, w którym możemy znaleźć w miarę obszerną instrukcję z kartą rejestracyjną, dyskietkę z programem oraz tekturową nakładkę na klawisze funkcyjne Amigi, która znakomicie ułatwia pracę z programem.

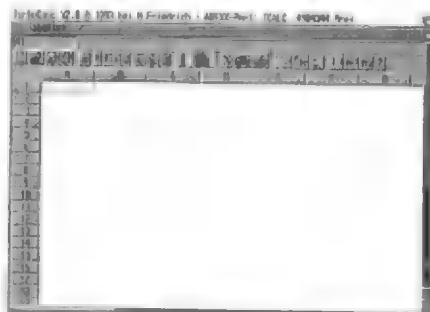
TurboCalc v 2.0 nie jest programem bardzo wymagającym. Można go z powodzeniem uruchomić na zwykłej Amidze wyposażonej w 512KB pamięci. Oczywiście trzeba się liczyć z mniejszą szybkością wyszukiwania danych (idealny jest tu twardy dysk) oraz z niemożnością tworzenia bardziej skomplikowanych tabel - do tego potrzeba przynajmniej 1MB pamięci. Instalacja programu przebiega w wyjątkowo nieskomplikowany sposób - po włożeniu dyskietki do stacji możemy program albo uruchomić, albo przeciągnąć jego szufladę do odpowiadającego nam miejsca na twardym dysku.

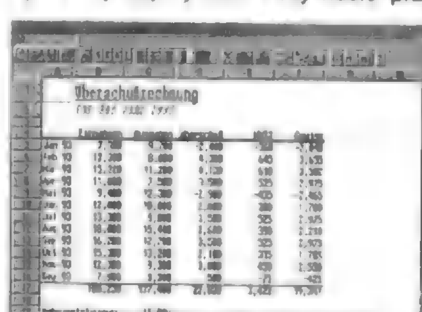
Po uruchomieniu programu pojawia się gotowy do pracy, dobrze zorganizowany ekran roboczy.

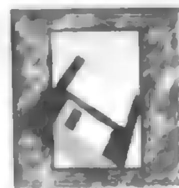
Możemy zacząć wpisywać nasze dane. Oczywiście na początku organizujemy tabelkę - podpisujemy ją w pionie i poziomie i możemy jechać.

Oczywiście tak zaczną pracę Ci, którzy zetknęli się już gdzieś z arkuszami kalkulacyjnymi. Pozostałych należy odesłać do instrukcji. Dostarczony manual jest wprawdzie w języku niemieckim, ale kto wie czy za jakiś czas nie ujrzy światła dziennego wersja polska. Instrukcja zaczyna się wprowadzeniem do TurboCalca, po którym następuje zwięźle napisany rozdział uczący co i jak z tym programem, czyli klawiszologia stosowana do obliczeń rzędu wyższego. Uczymy się tu co to jest komórka, jakie może przyjmować wartości. Przestaje być dla nas tajemnicą sprawa zaznaczania kilku komórek, przenoszenia ich z miejsca na miejsce i wstawiania tam gdzie akurat powinny być. Tu też dowiadujemy się jaka jest różnica pomiędzy formułą i formatowaniem i do czego może przydać się wiedza o tworzeniu bazy danych. Dodatkowo do tego rozdziału autorzy programu dołączyli specjalną szufladę z przykładami wziętymi wprost z książki tak, żebyśmy nie musieli tego wszystkiego ręcznie przepisywać. Oczywiście przy







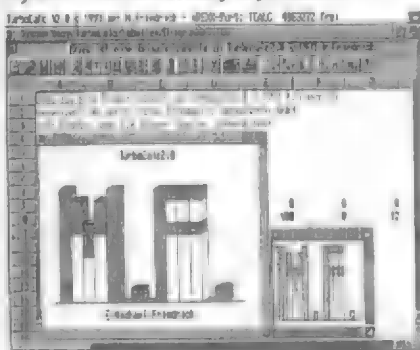


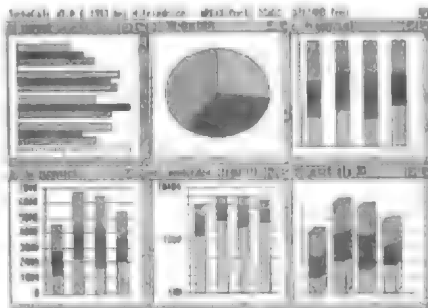
kłady z szuflady Tabellen można wykorzystać do zrozumienia dokładnych zasad działania programu, które są omówione w toku całego podręcznika. Pewnego rodzaju niedopatrzeniem, ale chyba trudnym do uniknięcia, wydaje się plik: Anzeigen_Tut.TCD, który pojawia się na pierwszym miejscu w szufladzie Tutorial. Każdy rasowy amigowiec zaraz go wywoła i wtedy... Oczywiście nie dzieje się nic strasznego. Po prostu program wchodzi w opcję wyświetlania wszystkiego po kolei. Krok po kroku przedstawiane są zmiany w przykładach. Efekt jest w pierwszym momencie taki, jakby coś się zawiesiło i tylko od czasu do czasu mrugają nam nowe tabelki. Paniczne wciskanie gadżetu zamknięcia okna nic tu nie pomaga. Program można zatrzymać jedynie opcją z menu Macro - Absp. stoppen. Tylko, że najpierw trzeba na to wpaść. Na wszelki wypadek radzę nie zgrywać tego co powstanie w toku działania wyświetlacza. Poza tym można spokojnie zresetować komputer. Chyba, że gdzieś w czeluściach RAMu mamy inne, ważne dane, które niechybnie ulegną skasowaniu.

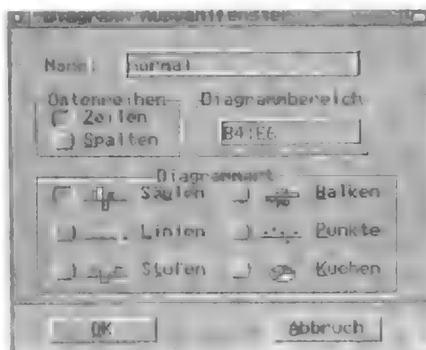
Bogato zaopatrzona została opcja ustawiania wyglądu programu. Możemy praktycznie wybrać każdy dostępny tryb graficzny, lecz przy włączeniu Interlace pojawi się na tyle uciążliwe migotanie ekranu, że można sobie albo dać spokój, albo dokupić kartę antydrganiową.

Program wprawdzie współpracuje również z nowymi trybami graficznymi wykorzystywanymi w komputerach Amiga 1200/4000, ale wyświetlane jest maksymalnie 8 kolorów, co wcale nie jest imponującą liczbą, nawet przy komputerach 8-bitowych. Do tworzenia prostych diagramów wystarczy to w zupełności, ale mogłoby być lepiej.

Diagramy tworzy się w sumie bardzo przyjemnie. Wystarczy zaznaczyć odpowiedni obszar arkusza kalkulacyjnego i pstryknąć na ikonkę rysowania diagramów z belki roboczej. Mamy tu do wyboru różne formaty wykresów:







Liczba możliwości nie jest wprawdzie imponująca, ale niektóre nowoczesne diagramy w rozwiniętych programach IBMowych są naprawdę niezwykle rzadko stosowane i brak przejrzystych możliwości wcale nie przynosi ujmy TurboCalcowi.

Jeżeli już jesteśmy przy wadach i grafice to koniecznie musimy wspomnieć, że program nie ma możliwości generowania grafik wektorowych - to co zostanie "wyprodukowane" przy jego pomocy może być dalej wykorzystywane, ale tylko jako grafika bitmapowa (ILBM). Oczywiście zasadniczą wadą takiej grafiki jest brak możliwości skalowania, co jest szczególnie widoczne, gdy chcemy ją wkomponować na stronę jakiegoś wydruku np. pracy magisterskiej. Wyjdzie ładnie, ale czasami dla przyzwoitego efektu będzie się trzeba nabiedzić. Niezbyt płynne jest też tworzenie napisów i praktycznie nie ma możliwości ich dowolnego ustawiania tam, gdzie by nam akurat najlepiej pasowały. Nie możemy się też popisać przy wydruku ponieważ nie ma możliwości jego podejrzenia na ekranie. Trzeba zmarnować nieco taśmy.

Program ma szeroki zakres wszelakiego rodzaju formułek stosowanych zwykle w arkuszach kalkulacyjnych. Za jednym naciśnięciem guzika otrzymujemy do wyboru ponad 100 różnych formułek, mogących bardzo ułatwić życie.

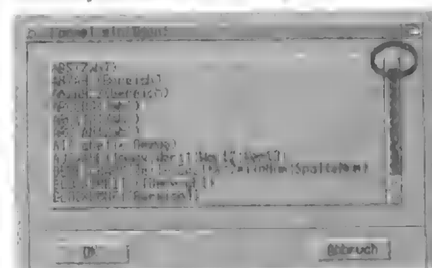
Zatem skalkulowanie dowolnego bilansu i przetworzenie go na język grafiki przestaje być w tym momencie problemem. Wszelkie "masowe" operacje podliczania i przeliczania ułatwia nam możliwość zastosowania ARexxu. Mamy do dyspozycji 3 gotowe programiki i opis poszczególnych funkcji, z których możemy sobie zaprogramować to, co jest nam akurat niezbędne. Łatwiej oczywiście tworzy się makropolecenia. Wystarczy włączyć nagrywanie, jak w zwykłym magnetofonie, a potem można wykonać to, co ma być wykonywane wiele razy. Z doświadczenia jednak wiem, że często zdarzają się małe pomyłki, czasami zaś chętnie zmienilibyśmy jakiś maleńki fragmencik makra. Wypada zatęsknić za programem MetaMac, który jest oferowny wraz z CEDem. Takiego programu jednak nie ma przy żadnym innym pakiecie, a miejmy nadzieję, że twórcy CEDa i tutaj przetrą innym ścieżkę.

Niezbyt imponująco przedstawia się opcja Help (Hilfe). Nie ma tu mowy o żadnym udziale użytkownika. Po prostu wybieramy odpowiednią opcję z menu i czytamy to, co nam się pojawi. Wprawdzie mamy tu do dyspozycji podstawowe informacje o programie, ale wszelkie szczegóły, które tak często stają się potrzebne w toku pracy musimy sami wyszukiwać z podręcznika.

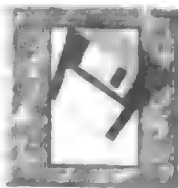
Podręcznik jest wprawdzie przyzwoicie napisany, ale można mu też przyczepić kilka łatek. Nie wiadomo, czy to na skutek błędu w druku, czy też zamierzenia twórców, tytuły poszczególnych rozdziałów i ważniejsze hasła zostały wydrukowane na szaro. Bardz źle się to czyta i wcale nie rzuca się w oczy.

Program pracuje w języku niemieckim, ale istnieje w nim możliwość tłumaczenia poszczególnych pojęć na język angielski dla osób, które przywykły pracować z arkuszami w tym języku. Niestety - angielskich opcji nie znajdzie się w indeksie, co nawet osobom znającym oba języki może bardzo utrudnić pracę.

Podsumowując możemy powiedzieć, że program jest dobry. Łatwość obsługi, szybkość działania i małe wymagania sprzętowe rekompensują w znacznym stopniu niedobory w sferze grafiki i dopracowania niektórych funkcji. Chociaż dla osób początkujących przydałaby się bardziej obszerna instrukcja. ❑



To tylko mała część możliwości formułowania...



VIDEO-TRACKER

Wojciech B. Lamer

Videotracker jest kolejnym po przeróżnych RSI Demo Makerach, Intro Makerach, itp., programem do tworzenia komputerowych teledysków zwanych potocznie demami lub jak kto woli demosami.

W Video Trackerze zastosowano trochę inny sposób tworzenia demonstracji niż w szeroko rozpowszechnionym RSI Demo Makerze. Mianowicie w RSI tworzyło się dema, w których efekty następowały płynnie po sobie, zaś Video Tracker pozwala na perfekcyjne wręcz zgranie danego efektu z muzyką. Dzięki temu możliwe jest zrobienie programu, który nadawałby się na dyskotekę (muzyka techno oraz migające efekty). Jak zapewnia autor programu - Peter van Campen - możliwe jest także wykonanie, dzięki zastosowaniu kolorowych czcionek i przesuwających się napisów, krótkich prezentacji. Co prawda nie mamy tu do wyboru tylu efektów co w Scali czy też Media Point, ale wymagania programu są o wiele mniejsze. Program współpracuje w pełni z GENLOCKIEM dzięki czemu coraz bardziej zbliża się do Scali. Posiada on jeszcze jedną dużą przewagę nad Scalą. Mianowicie kosztuje "tylko" 50 dolarów albo 100 marek niemieckich. Osobiście uważam, że program tego typu powinien być rozprowadzany bezpłatnie, ale dla innych cena ta może być interesująca.

Edytując naszą demonstrację możemy użyć obrazków we wszystkich trybach graficznych Amigi (brak na razie wersji działającej z trybami A1200 i A4000), animacje nagrane w formacie Anim5, wycinki obrazków (brushe), definicje obiektów wektorowych, fonty (także kolorowe) oraz palety kolorów. Jeżeli zaś chodzi o muzykę, program rozpoznaje format Sound Trackera. Nie wiadomo niestety czy działają wszystkie muzyczki napisane pod Pro Trackerem. Stworzone demo nagrywamy na dysku jako tzw. VIDULE, które możemy następnie spakować dowolnym packerem (np. Power Packerem). Wykreowane w ten sposób demko uruchamiamy podając w CLI jego nazwę, przy czym żaden inny plik nie jest potrzebny do poprawnego jego działania. Na Video Trackerze możemy stworzyć zarówno krótkie interko, jak i bardzo rozbudowane dema, których długość jest uzależniona jedynie od ilości wolnej pamięci.

W zestawie Video Tracker znajdują się dwie wersje tego programu:

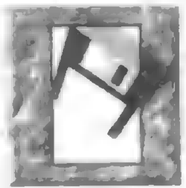
Video Tracker Guideline

Wersja programu działająca pod systemem, korzystająca z multitaskingu. Przeznaczona jest tylko dla użytkowników OS 2.0 lub wyższej. Najdziwniejsze jest to, że wersja ta nie chciała działać na Amidze 1200, na której program był testowany.











Video Tracker Interactive

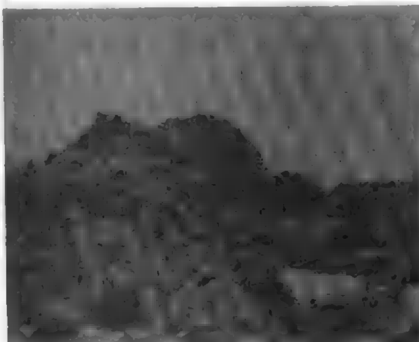
Jest to wersja działająca też na Kickstartach 1.3. Nie korzysta niestety z multitaskingu, a sam interface użytkownika nie został najszczęśliwiej zaprojektowany.

Dodatkowo na dysku znajdują się przykładowe projekty, grafiki, muzyczki, itp. oraz instrukcja programu.

Do Video Trackera powstało już bardzo wiele przykładów, jak np. animacje tańczących postaci, kroje liter, obrazki, efekty (popularna w wielu demach wektorówka czy też efekty punktowe) oraz przeróżne muzyczki i programiki pomocnicze. Można podać nazwy takich jak np.: Turbo Imploder 4.0 (najlepszy obecnie packer), Digital Illusions (program do obróbki grafiki), Scenery 1.0 (do tworzenia krajobrazów).

Dyski te można nabyć u autora programu, a jest ich w sumie 11:

VideoTracker Video Examples 1
Dwa krótkie demka w wersji na 512KB i 1MB pamięci.





VideoTracker Video Examples 2
Duże demo działające tylko na 2MB.

VideoTracker Video Examples 3
Demko z ciekawymi animacjami. Także tylko na 2MB

VideoTracker Video Examples 4
Demonstracje wektorowych kulek oraz innych obiektów wektorowych.

VideoTracker Scroll Examples
Dysk pomocny przy tworzeniu prezentacji zawierający przeróżne scrolle.

VideoTracker Object Examples
Procedury pozwalające na korzystanie we własnych demkach z przeróżnych obiektów.

VideoTracker Vector Examples
Znane z demosów efekty z grafiką wektorową.

VideoTracker Utilities 1
Dysk z przeróżnymi programikami pomocnymi w stworzeniu naszego dzieła, np. packery, obróbka grafiki, itp.





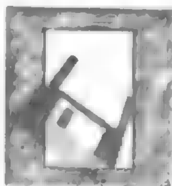
VideoTracker Utilities 2
Druga część poprzedniego dysku.

VideoTracker Vector-Objects 1
Dysk z obiektami wektorowymi, których możemy użyć we własnych produkcjach zawierających elementy grafiki wektorowej.

VideoTracker Developer Kit
Procedury w asemblerze z dokładnymi opisami. Dzięki nim możemy zaadoptować własne procedurki do Video Trackera. Szczerze mówiąc nie sądzę żeby ktoś, kto zna asembler na tyle by pisać własne efekty, był zainteresowany ich przerabianiem, a potem tworzeniem dema z wykorzystaniem Video Trackera.

Podsumowując, jest to program z ciekawymi opcjami, efektami, etc. Nie wydaje mi się jednak żeby cena 50 dolarów była ceną odpowiednią jak na program do robienia niezbyt wysokiej jakości programów nie dających żadnego wymiernego zysku (oprócz chwały), a właśnie takimi programami są dema.





AMOS Professional v2.0

[illegible]

Popularność języka AMOS BASIC przeszła nawet najśmielsze oczekiwania autorów. Programują w nim chętnie zarówno początkujący, jak i profesjonaliści. Pierwsi chwalą ogromną łatwość programowania, a drudzy duże możliwości pozwalające w znacznym stopniu wykorzystać sprzętowe właściwości komputerów AMIGA.

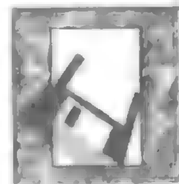
Po takim sukcesie autorzy nie spoczęli na laurach i podjęli pracę nad udoskonaleniem swojego produktu. Powstawały kolejne, coraz lepsze wersje, pojawił się kompilator i wydawało się, że rozwój Amosa będzie konsekwentny, lecz powolny. Przełomowym okazał się moment wypuszczenia pakietu AMOS Professional, który wyznaczył nowy standard tego języka.

Na początek trochę historii. Jeszcze kilka [illegible] temu Basic kojarzony był [illegible] prostym, wręcz przeznaczonym dla dzieci językiem programowania i komputerami Atari, Commodore 64, Spectrum, Amstrad itd. Miały one interpreter tego języka umieszczony w pamięci ROM, [illegible] możliwości jakie oferował były raczej skromne. Trudno teraz dociec jak to się stało, faktem jest, że głównie za sprawą "fanatyków" komputera IBM korzystających z Pascala czy C, Basic został "zgnębiony" i wyrzucony poza margines jako język powolny, niestrukturalny i ogólnie mało przydatny. Pogląd ten ugruntowała fatalna wręcz jakość dostępnych wtedy interpreterów. Także i AmigaBasic dołączony do oprogramowania systemowego Amigi prezentował poziom raczej niski i skutecznie zniechęcał "świeżo upieczonych" amigowców do Basica. Pojawienie się pierwszych wersji AMOS BASICa wywołało spore poruszenie. Jedni starym zwyczajem wyśmiewali Amosa jako kolejną mutację Basica, a inni wychwalali go jak tylko mogli. Mówiło się nawet o zmierzchu gier pisanych w asemblerze i świetlanej przyszłości rozrywkowego oprogramowania tworzonego w Amosie. Jak to zwykle bywa, wszyscy mieli trochę racji i prawda leżała pośrodku. Profesjonaliści nadal pisali gry w asemblerze, ale dla amatorów Amos był idealny. Pozwalał łatwo i szybko osiągnąć to, co poprzednio wymagało sporej pracy popartej fachową wiedzą. Zaczęły więc powstawać pierwsze komercyjne programy pisane w Amosie i język ten wszedł na stałe do grona cenionych i często używanych. Najnowsza jego wersja, czyli AMOS Professional V2.0 daje jeszcze większe możliwości niż poprzednia. Jeśli dodamy do tego spo[illegible] ułatwienia dla programisty jakie niesie ze sobą Amos Pro, to jest bardzo prawdopodobne, że za jakiś czas Amos ulokuje się w czołówce języków programowania dostępnych dla Amigi. Warto więc przyjrzeć mu się nieco bliżej.

Pierwsze, co rzuca się w oczy przy kontakcie z nowym interpreterem to fakt, że bardzo się "rozrósł". Mieści się bowiem [illegible] sześciu dyskietkach wypełnionych po brzegi. Ich zawartością zajmę się w dalszej części, [illegible] teraz chciałbym napisać kilka zdań na temat zmian w samym edytorze. W górnej części po dosyć dziwnym, choć ciekawym menu znanym z poprzednich wersji, pozostało jedynie kilka ikon pozwalających na uruchomienie najczęściej używanych opcji. Pozostałe funkcje zostały umieszczone w typowym menu, rozwijanym z górnej belki. Ilość opcji w nim zawartych robi duże wrażenie. Wszystko mamy ułożone w ośmiu grupach tematycznych.

I tak w menu Amos znajdziemy oprócz typowych funkcji About i Quit, bardzo ciekawą możliwość ładowania programów pomocniczych (Accessory). Są to w gruncie rzeczy zwyczajne programy napisane w Amosie. Po wczytaniu są przechowywane w pamięci, [illegible] w menu Amos pojawiają się dodatkowe opcje (tytuły tych programów), po wybraniu których można je uruchomić. Pozwala to użytkownikom dysponującym odpowiednią ilością wolnej pamięci na dołączenie do edytora często wykorzystywanych programów np. konwertujących dane graficzne, czy muzyczne i używanie ich tak jak każdej innej opcji edytora. Jest to bardzo wygodne, choć jak już wspomniałem wymaga rozszerzonej pamięci. Standardowy 1MB może okazać się zatem ilością zbyt małą do wykorzystania tej możliwości edytora.

Menu Project oferuje opcje podstawowe, takie jak uruchamianie napisanego programu, testowanie, wczytywanie i zapisywanie kodu źródłowego na dysk itd. Na uwagę zasługują opcje Open New i Open & Load pozwalające na pisanie bądź wczytanie tekstu programu w nowym oknie. Jest to nowość w stosunku do poprzedniej wersji dająca możliwość pracy [illegible] kilku tekstach równocześnie. Są to możliwości często spotykane w edytorach tekstu i naprawdę brakowało ich w poprzednich wersjach edytora Amosa. System taki pozwala na swobodne pisanie w nowym oknie procedur próbnych i jeśli



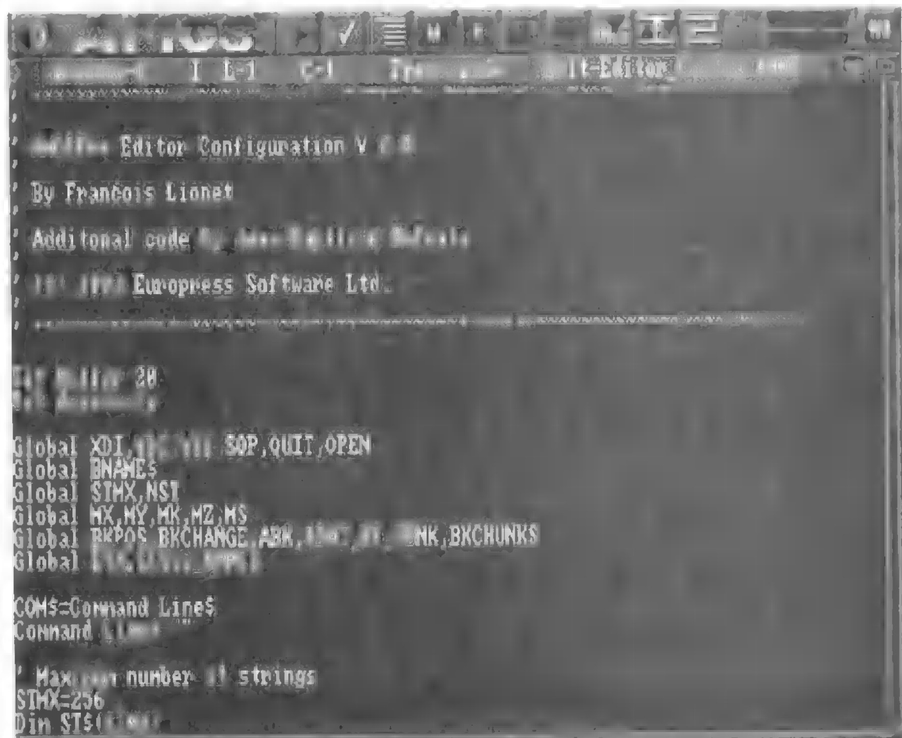
ich działanie spełni nasze oczekiwania, możemy przenieść je do okna zawierającego program główny. Nowym, i ciekawym elementem jest funkcja Monitor. Po jej wybraniu pojawia się nowy ekran stanowiący pewnego rodzaju edytor przeznaczony do krokowego uruchamiania programów. Przenoszony jest do niego automatycznie listing znajdujący się w edytorze głównym. Podstawowym zastosowaniem takiego rozwiązania jest wyszukiwanie błędów w programie. W lewym górnym rogu mamy okienko, w którym możemy oglądać wyniki działania naszego programu, poniżej widoczny jest listing programu z oznaczoną aktualnie wykonywaną komendą. Prawy górny róg, to panel kontrolny pozwalający sterować całością za pomocą myszki. Całości obrazu dopełnia umieszczone na dolnej krawędzi okno komunikacyjne, w którym podawane są komunikaty. Takie rozmieszczenie elementów na ekranie pozwala nam bardzo wygodną i łatwą obsługę. Funkcja Monitor jest szczególnie przydatna przy wyszukiwaniu błędów w bardziej skomplikowanych programach, zawierających dużo procedur i pętli warunkowych. Przy jej pomocy można szybko prześledzić działanie programu, porównując efekty jego działania widoczne na pomniejszonym ekranie ze znajdującym się poniżej listingiem. Zlokalizowanie nieprawidłowo działającej instrukcji jest zatem proste i co najważniejsze szybkie. Poza wywołaniem Monitora i wspomnianymi już instrukcjami w menu Project znajdziemy jeszcze funkcję Check 1.3 testującą program pod względem zgodności z Amosem w wersji 1.3. Można więc szybko dowiedzieć się, czy właśnie napisany program będzie działał po wczytaniu do edytora starszej wersji Amosa i czy da się skompilować poprzednim kompilatorem.

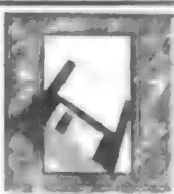
W menu Editor znajdziemy szereg opcji sterujących pracą edytora. Większość z nich znana jest z poprzednich wersji (np. zwijanie i rozwijanie procedur). Dołączono natomiast kilka funkcji sterujących pracą z oknami i bardzo rozbudowano możliwości przemieszczania się po programie. Możemy wykonywać skoki do wybranej linii, początku tekstu, jego końca, itd. Każdorazowe wybieranie opcji z menu jest raczej mało wygodne przy takich operacjach, lecz po opanowaniu odpowiednich kombinacji klawiszy, te mocno rozszerzone możliwości mogą znacznie przyspieszyć pracę. Mamy możliwość umieszczania w tekście znaczników i wykonywania skoków do miejsca ich ustawienia oraz kasowania tekstu na wiele sposobów (całą linię, wszystko na lewo od kursora, wszystko na prawo od kursora, itd.). Pomocna może się okazać możliwość wprowadzania makrodefinicji. Przyporządkowanie określonej kombinacji klawiszy całego ciągu operacji jest funkcją bardzo pożądaną w każdym programie, a szczególnie podczas programowania. Wszystkim roztargnionym polecam dwie ostatnie opcje z tego menu, czyli Undo i Redo. Mogą one czasami "wyciągnąć" programistę z naprawdę trudnej sytuacji.

Menu Block oferuje zestaw funkcji bardzo zbliżony do znanego z poprzednich wersji. Są to operacje standardowe dla każdego edytora tekstu i w tej dziedzinie niewiele nowego da się wymyśleć. Możemy wycinać fragmenty tekstu, przenosić je w inne miejsce, nagrać na dyskietkę, wydrukować, itd. Jest to moim zdaniem wystarczający zestaw operacji, pozwala bowiem na swobodne operowanie blokami tekstu i w praktyce spisuje się doskonale.

Podobnie ma się sprawa z funkcjami poszukiwania i zamiany ciągu znaków, umieszczonymi w menu Search. W stosunku do poprzednich wersji jest tu niewiele zmian. To co mamy do dyspozycji, podobnie jak dawniej, spisuje się bardzo dobrze.

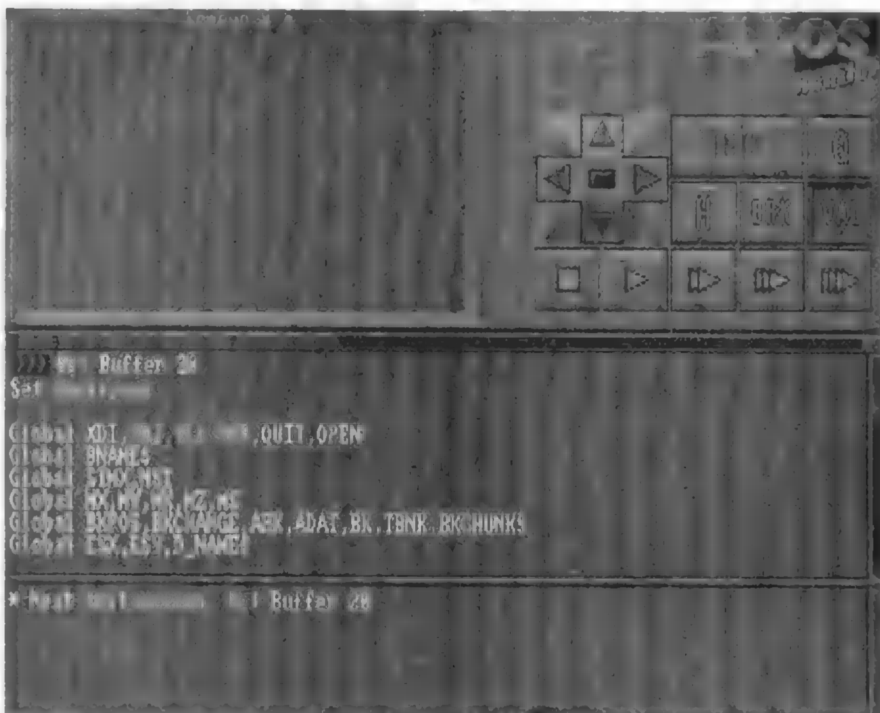
Za to w menu Config możemy zaobserwować zmiany wręcz rewolucyjne. Dostępnych jest szereg opcji, przy pomocy których można wprowadzić wiele zmian do pracy całego Amosa. W zasadzie w ręce użytkownika przekazana została pełna kontrola nad pracą systemu. Zmieniać można praktycznie wszystko. Nie będę opisywał wszystkich możliwości, gdyż zajęłoby to bardzo dużo miejsca i pewnie byłoby trochę nudne. Wspomnę tylko, że ustawianiem parametrów sterują dwa, dosyć rozbudowane programy napisane w Amosie, oddzielne dla parametrów edytora i interpretera. Wywoływane są one i doczytywane z dyskietki po wybraniu poszczególnych opcji w menu Config. Szeroki dostęp do "wnętrza" Amosa uważam za rzecz bardzo przydatną i ciekawą. Przy odrobinie cierpliwości użytkownik może dostosować cały system co swoich potrzeb. Odpowiednio zmienić rozłożenie opcji w menu, kombinacje klawiszy, a nawet komunikaty podawane podczas pracy. Nowy Amos jest więc zgodny z obecnym kierunkiem w rozwoju wszelkiego oprogramowania. Można go z całą pewnością uznać za program przyjazny dla użytkownika i potrafiący dostosować się do jego wymagań.





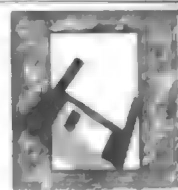
Na potwierdzenie tej tezy przedstawię kolejne menu, User. Standardowo umieszczone są w nim tytuły niektórych programów pomocniczych, dostarczonych wraz z Amosem. Wybranie jednej z tych opcji powoduje wczytanie z dysku i uruchomienie przyporządkowanego jej programu. Opcja Add Option pozwala na łatwe umieszczenie w tym menu nowych opcji i przyporządkowanie im innych, np. własnych programów. Umożliwia to swobodne rozszerzanie możliwości edytora o dodatkowe funkcje za pomocą prostych, pisanych [redacted] Amosie programów. W przeciwieństwie do programów pracujących jako Accessory, są [redacted] każdorazowo doczytywane z dysku, co może mieć spore znaczenie dla wszystkich dysponujących małą ilością pamięci.

Jeśli ktoś się pogubi w tej masie komend i funkcji, może skorzystać z menu Help. Nad funkcją pomocy dla użytkownika chciałbym zatrzymać się nieco dłużej, gdyż jest ona wykonana doskonale i perfekcyjnie spełnia swoje zadanie. Wszystkie instrukcje Amosa zostały pogrupowane w bloki tematyczne wywoływane przez wybranie odpowiedniej opcji z menu Help. Po jej uruchomieniu na ekranie pojawia się nowe okienko ze spisem wszystkich związanych z tym tematem instrukcji. Kliknięcie [redacted] jednej z nich wywołuje na ekran naprawdę dokładny opis wybranej komendy. Jeśli jednak opis będzie niewystarczający, możemy jednym ruchem myszką wywołać na monitor przykładowy programik demonstrujący interesującą nas instrukcję. Jego uruchomienie i krótka analiza kodu pozwala zazwyczaj dokładnie poznać wszystkie aspekty używania potrzebnej nam komendy. System działa naprawdę sprawnie i uwalnia programistę od ciągłego wertowania podręcznika w przypadku jakichkolwiek niepowodzeń. Ma [redacted] spore znaczenie, bowiem AMOS Professional w stosunku do swych poprzedników został wyposażony w wiele nowych komend, których działanie z oczywistych względów jest słabo znane osobom programującym w poprzednich wersjach. Nieocenione usługi oddają tu wspomniane wcześniej programy przykładowe. Pozwalają one poznać nowe instrukcje i później stosować je we własnych programach, do czego nie będzie trzeba nikogo zachęcać, gdyż instrukcje te w zdecydowanej większości ułatwiają programowanie. Jako przykład podam możliwość bezpośredniego wykorzystywania modułów muzycznych w formacie Noisetrackera i kilka komend znacznie ułatwiających programowanie komunikacji z użytkownikiem (gotowe gadżety). Nowe możliwości Amosa to temat raczej do kursu programowania, [redacted] tutaj wspomnę tylko, że autorzy położyli główny nacisk na ułatwienia w programowaniu.



Edytor jest moim zdaniem o klasę lepszy od poprzedniego i zbliżony obsługą do popularnych edytorów tekstu, co ułatwia jego sprawne używanie. Muszę zaznaczyć, że amigowcy nie posiadający twardego dysku mogą trochę się rozczarować. Procedury pomocnicze, teksty do funkcji "help" czy programy przykładowe rozmieszczone są na różnych dyskietkach i w przypadku pracy [redacted] stacją dysków wspaniałe działanie niektórych opcji edytota, którym się tak zachwycałem może zostać zmącone sporą "dyskoteką". Jestem jednak dobrej myśli, gdyż [redacted] posiadaczy twardych dysków rośnie szybko, a dodatkową stację dysków, częściowo rozwiązującą wspomniany problem, posiada już bardzo wielu.

Zgodnie z zapowiedzią na początku artykułu, postaram się teraz przedstawić co znajduje się oprócz edytora na tych sześciu dyskietkach, jakie pakiet AMOS Professional zajmuje. Autorzy zadbali o to, aby przy programowaniu nikomu niczego nie zabrakło. Mamy więc sporo różnych programów ułatwiających pracę. Pozwalają one na konwersję standardu potrzebnych danych, szybkie stworzenie banku sampli, wycięcie [redacted] danego obrazka fragmentu i stworzenie z niego boba bądź sprite'a, edycję fontów, czy też kopiowanie plików. Jak można zauważyć rozpiętość tematyczna tych programów jest spora i nie sposób wymienić wszystkich. Oprócz tych, mających praktyczne zastosowanie programików, na dyskietkach można znaleźć całą masę procedur czysto przykładowych, prezentujących możliwości Amosa lub poszczególne komendy. Wszystko to wzbogacone zostało sporą ilością różnego rodzaju grafik, animacji, modułów muzycznych i gotowych banków pamięci zawierających tego rodzaju dane. Nie zabrakło miejsca dla przykładowych gier. Nie są one może wykonane na najwyższym poziomie, ale jestem pewien, że nie powstydziłby się takiego dzieła niejeden zawodowy programista. Zarówno gry, jak i pozostałe programy są napisane niezwykle przejrzyście, [redacted] umieszczone bezpośrednio w programach (w formie komentarzy) informacje dodatkowe zazwyczaj rozwiewają wszelkie pojawiające się wątpliwości. Wszystko [redacted] daje programiście spory komfort pracy. Gdy tylko trafi się jakiś problem, wystarczy przyjrzeć się odpowiednim programom przykładowym i podpatrzeć "jak zrobili to inni".





Wiem z własnego doświadczenia, że jest to naprawdę pomocne, gdyż pozwala częściowo ominąć niezwykle czasochłonny etap uczenia się na własnych błędach. Wszystkie te dodatki oceniam więc bardzo wysoko i radzę mieć dyskietki z nimi zawsze pod ręką, aby móc z nich skorzystać w wypadku pojawienia się trudności.

Wraz z nowym edytorem i interpreterem Amosa pojawił się także stosowny kompilator. Pierwszą różnicą jaką zauważa się w stosunku do poprzedniej wersji jest całkowicie nowa szata graficzna. Jest ona znacznie ładniejsza, a przy tym wszystkie opcje ułożone są o wiele wygodniej niż poprzednio, chociaż możliwości oferowa przez kompilator niewiele się zmieniły. Mamy do wyboru trzy możliwe warianty kompilacji (program do uruchomienia z CLI, Workbencha bądź edytora Amosa) oraz możliwość przeprowadzenia automatycznej kompresji skompilowanego programu. Jest ona tym razem bardzo wydajna i czasami daje lepsze wyniki niż użycie renomowanego programu Power Packer. Niewielkie zmiany dotyczą opcji dostępnych w preferencjach samego kompilatora jak i kompilowanego programu. Pewną innowacją jest możliwość dołączenia, bądź nie, biblioteki amos.library. Jeśli nie dołączymy jej do programu, podczas jego uruchamiania, będzie ona poszukiwana w katalogu LIBS dysku systemowego. Może to dać pewne oszczędności w ilości zajętego miejsca na dysku, jeśli będzie tam kilka programów napisanych w Amosie. Podstawową zmianą jaka wprowadzona została w kompilatorze jest możliwość kompilacji programów napisanych z wykorzystaniem nowych instrukcji zawartych w interpreterze. Kompilator podobnie jak interpreter został wyposażony w bardzo rozbudowany i zarazem pomocny "Help" oraz dużo umieszczonych na dodatkowym dysku przykładowych programów gotowych do kompilacji. Są w nich umieszczone różnego rodzaju liczniki wskazujące czas wykonania zadanych operacji i można bez trudu zobaczyć "przyspieszenie" jakie daje kompilacja programu.

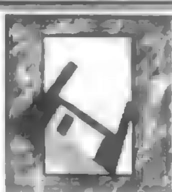
Ze względu na pełną zgodność interpretera i kompilatora stanowią one swego rodzaju zestaw, ściśle ze sobą współpracujący i pozwalający na tworzenie w Amosie profesjonalnych programów. Tak jak pierwsze wersje Amos Basica stworzone zostały głównie z myślą o pisaniu gier, tak AMOS Professional V2.0 może z powodzeniem służyć do stworzenia każdego rodzaju programu. Podstawową jego zaletą jest łatwość programowania i szybkość tego procesu. Zawiera on bowiem szereg instrukcji ułatwiających programowanie komunikacji z użytkownikiem i różnymi urządzeniami systemu. Zestaw dostępnych komend jest bardzo bogaty i pozwala w zasadzie na wykorzystanie wszystkich algorytmów działania programu. Moim zdaniem jedną z istotniejszych, ujemnych cech Amosa jest przeciętna prędkość działania. Wiem, że jest ona i tak o wiele wyższa niż innych Basiców (zwłaszcza po kompilacji), ale w niektórych przypadkach okazuje się niewystarczająca. Problem ten można rozwiązać, stosując procedury pisane w języku maszynowym, które bez trudu dołącza się do programu w Amosie. Zastępując obliczeniowe fragmenty programu procedurami napisanymi w asemblerze można uzyskać naprawdę dużą prędkość działania, która w połączeniu z bardzo dobrym interfejsem użytkownika stworzonym w Amosie może zaowocować programem najwyższej klasy. Myślę więc, że już niedługo ujrzymy szereg doskonałych programów, zarówno użytkowych jak i gier powstałych z wykorzystaniem Amosa, który w tej chwili niezaprzeczalnie wiedzie prym wsród Basiców na Amigę. Jeśli będzie on nadal udoskonalany, to z pewnością długo nie straci swego przewodniego charakteru. ❑

RGB SC
ul. Św. Ducha 7
33-100 Tarnów
tel. (014) 213762 po godz. 17.00

prowadzimy sprzedaż wysyłkową
•Komputery AMIGA4000/1200/CD32/2000 - DZWOŃ
•A570 CD Interface do A500 - 3.990.000 zł
•Karty RAM do A1200/2000/600/500 - DZWOŃ
•HDD 40-540MB 2.5, 3.5"AT/SCSI - DZWOŃ
•Kontrolery do A500 AT/SCSI - DZWOŃ
•Rewelacyjną kartę turbo do A1200!!!
TURBO-JET MC68030/MMU/FPU68881/28MHz - DZWOŃ
•Karty grafiki MERLIN i PICASSOII - DZWOŃ
•Fantastyczny digitizer VLab (SVHS) - DZWOŃ
•Stacje dysków HD 1.76MB - 4.300.000 zł
•NOWOŚĆ! 2MB CHIP RAM adpt. A500/2000 - 4.700.000 zł
•Części i podzespoły do komputerów Amiga
•PROWADZIMY SERWIS A500/2000 i C64

Po szczegółowy katalog i wykaz cen zadzwoń lub napisz !

Lublin
ul. Okopowa 6
tel. (0-81) 213-94
fax (0-81) 418-92
AMIGA
Komputery Literatura Monitory Programy Dyskietki Samplery Rozszerzenia Stacje dysków HD
Myszki Koprocesory Kontrolery Joysticki Genloki Skanery Kable Midi CD
już wiesz gdzie tego szukać ...



NIE TYLKO CED

Adam Drugowicz

W czasach, kiedy niepodzielnie zdaje się królować CED, zapominamy o podobnych edytorach tekstu należących do tej samej klasy. A przecież istnieją inne programy, które wcale nie są gorsze. Aby nie być gołosłownym przedstawię dwa z nich.

Na początek zajmę się programem o nazwie "EdWord". EdWord jest programem shareware, chociaż bezpłatnie możemy używać tylko wersji demonstracyjnej. Aby otrzymać pełną wersję tego edytora tekstu, należy wysłać odpowiednią kwotę do jego autora - Martina Reddy.

EdWord uruchamia się albo na własnym ekranie, albo jako okienko na ekranie Workbencha (można to ustalić przy pomocy funkcji "Screen mode" w menu "Tools"). W przypadku, gdy wybraliśmy pracę na ekranie Workbencha, zamiast używania opcji "Open" do wczytania tekstu możemy po prostu chwycić ikonkę pliku tekstowego i włożyć ją do okna EdWorda (użytkownicy Kickstartu 1.3 niestety nie mają tej możliwości), a tekst zostanie automatycznie wczytany do edytora, co czasami dość znacznie ułatwia pracę.

EdWord jest reklamowany jako edytor dla programistów, co nie jest stwierdzeniem na wyrost. Oferuje on kilka zaskakujących możliwości nie spotykanych dotychczas w innych edytorach. Po raz pierwszy możemy zdziwić się, próbując wczytać tekst spakowany Power Packerem. EdWord automatycznie rozpoznaje i rozpakowuje taki plik! To niby nic nie znacząca, ale czasami bardzo przydatna możliwość, a żaden z pozostałych programów do edycji tekstu jej nie posiada. Inną egzotyczną funkcją jest wbudowany kalkulator.

Nie jest on co prawda zbytnio rozbudowany, ale umożliwia przeprowadzenie podstawowych działań nie tylko na liczbach dziesiętnych, lecz również na szesnastkowych, binarnych i ósemkowych. Zalety tego narzędzia z pewnością docenią wszyscy używający EdWorda do pisania programów. Kolejną rzeczą, która także zachwyci programistów jest funkcja TextCasing. Umożliwia ona wykrywanie przez edytor wyrazów będących słowami kluczowymi wybranego języka programowania i automatyczną zamianę ich pisowni na małe lub duże litery (w zależności od tego jak sobie zażyczymy). Po wybraniu tej funkcji z menu "Utilities" otwiera się okienko, gdzie w polu "Keywords file:" wpisujemy nazwę pliku konfiguracyjnego dla danego języka programowania (klikając w gadżet LOAD otwieramy requester, w którym możemy wybrać ten plik w bardziej wygodny sposób). Gadżetami "Keyword Casing" i "Variable Casing" ustawiamy odpowiednio sposób pisowni słów kluczowych i nazw zmiennych w tekście naszego programu. Dostępnymi ustawieniami są:

OFF - EdWord nie będzie zmieniał wpisanych przez nas słów.

UPPERCASE - EdWord zamieni wszystkie litery na wielkie.

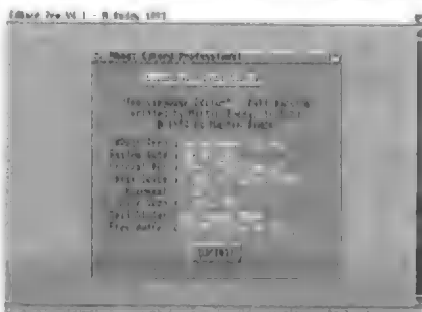
LOWERCASE - wszystkie litery będą małe.

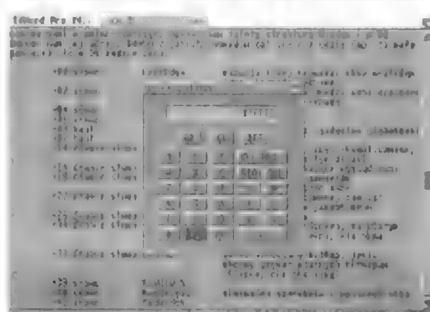
CAPITALISE - piersza litera będzie wielka, pozostałe małe.

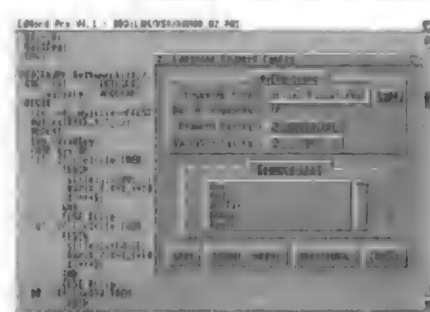
FROM FILE - edytor ustawi taką pisownię jak w pliku konfiguracyjnym.

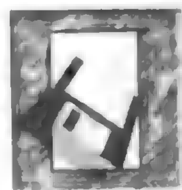
Poniżej w okienku z tytułem "Keyword List" możemy obejrzeć wszystkie słowa kluczowe z aktualnie wczytanego pliku konfiguracyjnego.

Oprócz tego możliwe jest ustawienie samoczynnego wcinania tekstu po napotkaniu w tekście określonych wyrazów. Np. dla Pascala będzie to BEGIN (dla wysunięcia tekstu w prawo) i END (dla wsunięcia tekstu w lewo). Klikając w pole "SYMBOL INDENT" możemy dowolnie wybrać słowa powodujące automatyczne formatowanie tekstu. Kolejną opcją przydatną dla piszących programy jest funkcja "Match brackets" z menu "Search". Sprawdza ona ilość otwartych i zamkniętych nawiasów. Wystarczy najechać kursorem



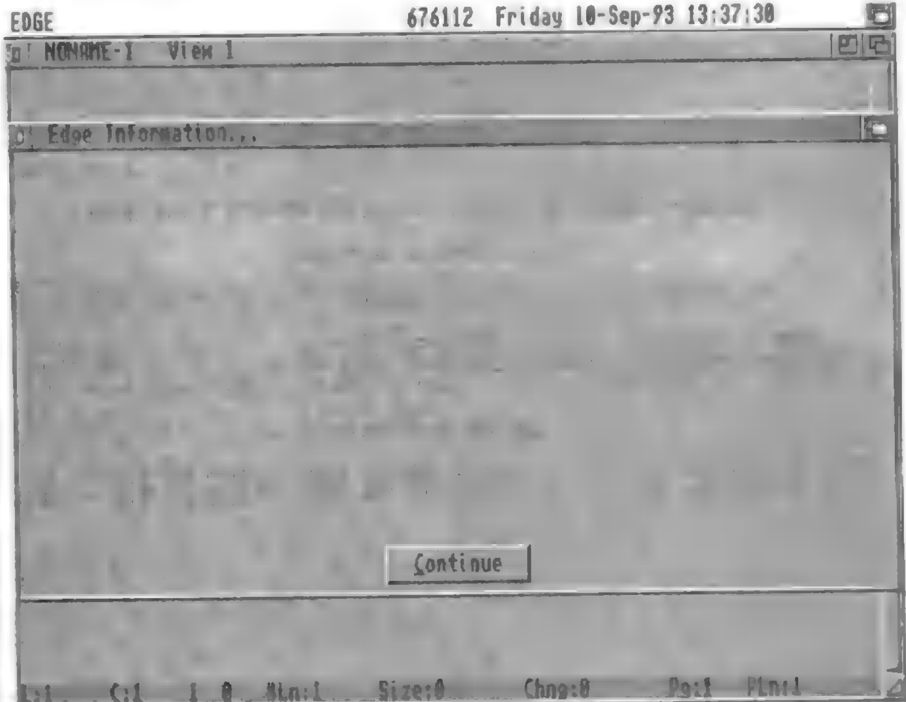






na początek wyrażenia matematycznego zawierającego nawiasy i wybrać "Match brackets", aby się dowiedzieć, czy zamknęliśmy tyle samo nawiasów co otworzyliśmy. Bardzo szybko można przekonać się, że przy przy bardzo długich i skomplikowanych wyrażeniach jest to niezastąpione. Następną ciekawostką jest możliwość uzyskiwania znaków o dowolnych kodach ASCII nie przez bezpośrednie wciśnięcia klawisza z danym znakiem (co przy różnych niestandardowych znakach może trwać latami), lecz w całkowicie inny sposób. Wystarczy wcisnąć lewy klawisz Amiga i trzymając go wpisać kod ASCII wymaganego znaku. Gdy puścimy klawisz Amiga znak ukaże się na ekranie. Ten sposób wpisywania działa we wszystkich requesterach EdWorda, tak więc można łatwo wyszukiwać wszystkie dziwaczne znaki. Oczywiście nie każdy zna kody ASCII wszystkich znaków. Dla tych osób jest funkcja "Insert Ascii" z menu "Utilities". Po jej uaktywnieniu otwiera się duże okno z wszystkimi znakami i ich kodami ASCII. Wystarcza kliknąć na wybraną literę, a zostanie ona wstawiona w tekst (jest to fantastyczne ułatwienie przy wpisywaniu tekstów obcojęzycznych z właściwymi danym językom znakami diakrytycznymi). Dodam, że CED nie ma tak wygodnego wstawiania znaków niedostępnych z klawiatury. Dosyć interesujące jest menu "Files", w którym można znaleźć funkcje kasowania pliku na dysku, zmiany jego nazwy lub jego bitów protekcji.

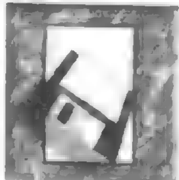
Oczywiście oprócz tych dosyć wyrafinowanych funkcji EdWord posiada typowe możliwości spotykane w edytorach tej klasy. Umożliwia więc zdefiniowanie trzech tak zwanych "bookmarków" - są to miejsca w tekście, do których można przeskoczyć jednym wciśnięciem klawisza. W programie EdWord te charakterystyczne miejsca definiujemy wciskając klawisz SHIFT i razem z nim jeden z klawiszy od F1 do F3, a skaczemy do nich wciskając SHIFT i F6 do F8. Trzy "bookmarki" przy dłuższym tekście to stanowczo za mało, ale osławiony CED również umożliwia zdefiniowanie tylko trzech takich punktów. Jak większość z ostatnio ukazujących się edytorów, EdWord udostępnia pracę na kilku tekstach jednocześnie, ich liczba jest jednak ograniczona do piętnastu. Niestety na ekranie możemy mieć jednocześnie tylko dwa dokumenty (używamy funkcji "Split screen" z menu "Tools"), ale za to możemy w bardzo wygodny sposób wybrać je spośród innych uaktywniając funkję "Select Doc" z menu "Project". Otwiera się wtedy okienko z listą dokumentów znajdujących się w pamięci. Wystarczy tylko kliknąć na nazwę interesującego nas tekstu i natychmiast pojawi się on na ekranie. Oczywiście EdWord udostępnia naturalne dla edytorów tekstu funkcje, jak wyszukiwanie i zamiana tekstu (z wyszukiwaniem do przodu lub do tyłu), wycinanie i kopiowanie fragmentów tekstu oraz standardowy już ostatnio interfejs ARexxa z ponad stu rozkazami.

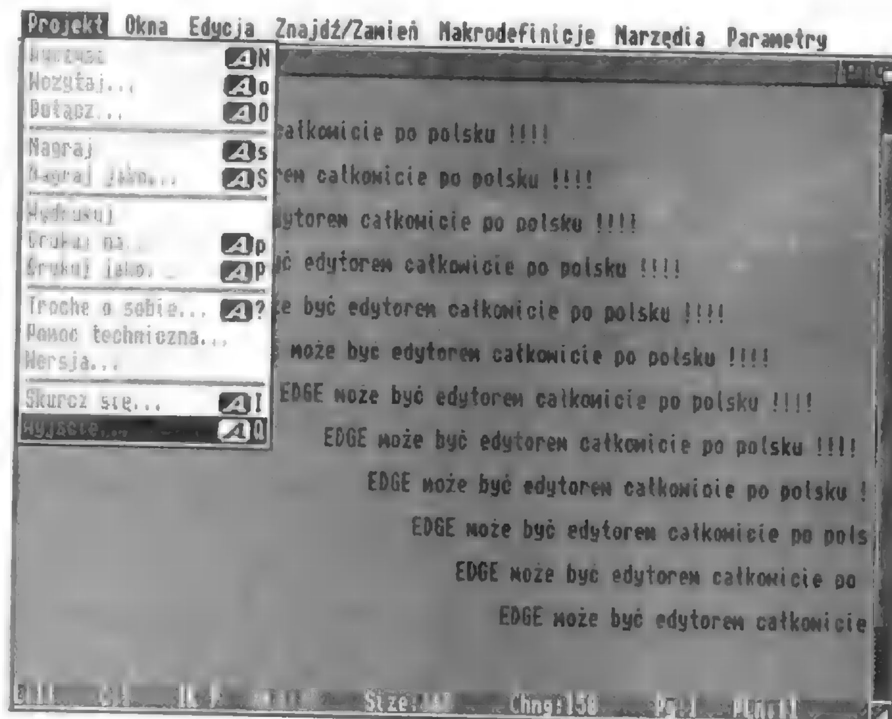


Po tych wszystkich pochwałach EdWorda czas na wytknięcie wad i niedogodności tego programu.

Pierwszą rzeczą, którą natychmiast się zauważa, jest zupełnie niestandardowa "klawiszologia". Tak dziwnych kombinacji klawiszy nie spotkałem do tej pory w żadnym edytorze. Na przykład aby rozpocząć zaznaczanie bloku tekstu, należy wcisnąć klawisz Amiga razem z przecinkiem, a żeby zakończyć zaznaczanie bloku należy wcisnąć Amiga razem z kropką. Co gorsze podczas zaznaczania bloku nie widać, aby coś się zaznaczało (we wszystkich innych edytorach zaznaczany tekst zmienia kolor na inny). Dopiero po wciśnięciu kombinacji Amiga+. (lub wybraniu funkcji "Block End" z menu "Edit") możemy zobaczyć co właściwie zaznaczyliśmy (wybrany tekst zmieni kolor). Inną denerwującą właściwością EdWorda jest bardzo mało pomocne Undo. Otóż Undo działa tylko na aktualnie zmienianej linii, jeśli przejdziemy do innej linii (wciskając RETURN lub przy pomocy kursorów), nie uda nam się już odtworzyć zawartości linii poprzedniej. Tak samo, gdy użyjemy funkcji "Delete line" (Amiga+Y), możemy sobie używać funkcji Undo do woli. Jedynym sposobem odtworzenia tak skasowanej linii jest wybranie "Undelete line" (Amiga+U). Są to wady raczej poważne, skutecznie utrudniające pracę z programem.

Poza wszystkimi zaletami i wadami EdWord posiada możliwość ustawienia wielu parametrów pracy. Po wybraniu funkcji "Preferences" z menu "Utilities" możemy włączyć (lub wyłączyć) wiele opcji jak np. nagrywanie ikonek, tworzenie plików .BACKUP, automatyczne przenoszenie wyrazów po dojściu do końca linii, mruganie kursora lub rodzaj używanych requesterów (możemy wybrać requester z biblioteki asl lub requester wbudowany w program). Oczywiście możemy także wybrać tryb ekranu, na którym chcemy pracować (chociaż EdWord nie obsługuje najnowszych trybów AGA). Możemy również ustalić, który kolor z palety będzie używany do wyświetlania danego elementu (np. tła, tekstu lub ramek). Jako ciekawostkę podam, że podczas używania EdWorda można się czasami natknąć na dowody poczucia humoru jego autora (wystarczy wcisnąć "CTRL+", "CTRL+]", "SHIFT+F5", "SHIFT+F9", "SHIFT+F10") na szczęście nieszkodliwe dla wpisywanego przez nas tekstu.





Innym edytorem tekstu należącym do tej samej grupy co CED jest Edge. Edge jest wydawany przez firmę INOVAtrovics, znaną choćby z doskonałego Directory Opusa.

Tak samo jak EdWorda możemy Edge uruchomić jako okno na ekranie Workbencha lub na oddzielnym ekranie. I identycznie jak we wcześniej opisywanym programie, w przypadku uruchomienia Edge jako okna na ekranie Workbencha, możemy wczytywać teksty do edytora przenosząc ikonkę pliku tekstowego do wnętrza okna Edge. Podobny mechanizm istnieje również, gdy Edge pracuje na własnym ekranie. W tym przypadku na ekranie Workbencha pojawia się ikona dysku o nazwie "Edge-Docs". Możemy ją otworzyć jak ikonkę zwykłego dysku i umieszczając tam ikony plików z tekstem powodować ich wczytanie do edytora. Podobną możliwość udostępniały do tej pory tylko programy edytorów należących do klasy wordprocesorów (na przykład ProWrite lub WordWorth). Jednak nie ma róży bez kolców. Aby móc używać większości funkcji Edge, musimy mieć uruchomionego ARexxa. Bez niego nie działa wiele podstawowych funkcji częstując nas komunikatem "No rexxmaster active", ponieważ program działa tylko na komputerach wyposażonych w kickstart 2.0 lub wyższy. Nie powinno to sprawić żadnych problemów, gdyż ARexx jest dołączony do tych systemów. Dla zupełnie nieznających się na ARexxie podam, że aby go zaktywizować należy uruchomić programik RexxMast (dbając jednocześnie, aby w katalogu LIBS była biblioteka rexxsyslib.library). Jeśli uda nam się uruchomić ARexxa, możemy korzystać ze wszystkich możliwości Edge.

Edge umożliwia pracę na nieograniczonej ilości dokumentów (nieograniczonej należy czytać: takiej, na ile starczy pamięci). Każdy z nich możemy oglądać w oknie dowolnej wielkości, więc oczywiste jest, że przy większej ilości tekstów po pewnym czasie na ekranie robi się bałagan. Lecz Edge umie sobie z nim poradzić. W menu "Windows/All windows" znajdują się trzy funkcje:

To maximum size - ustawia rozmiar wszystkich okien na maksimum. Wszystkie okna leżą jakby jedno na drugim a dostęp do kolenych uzyskuje się klikając w gadżet zmiany głębokości w prawym górnym rogu okna.

Panel vertically - wszystkie okna pojawiają się jedno obok drugiego.

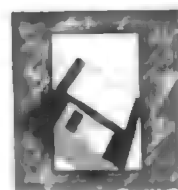
Panel horizontally - wszystkie okna wyświetlane są jedno pod drugim.

Jednocześnie, gdy wybierzemy funkcję "General..." z menu "Settings", a następnie klikniemy w pole "Windows", możemy ustalić, że po kliknięciu myszką w jakieś okno z tekstem, pojawia się ono na pierwszym planie - wystarczy zaznaczyć opcję "AutoFront". Oprócz tego po zaktywizowaniu okienko może powiększać się do maksymalnych rozmiarów (opcja "AutoZoom"), a po deaktywacji wracać do poprzednich rozmiarów (opcja "AutoUnZoom").

Z pracą na wielu oknach związana jest jeszcze jedna, nie spotykana dotychczas opcja. Wybierając z menu "Windows" funkcję "Split window" uzyskujemy "rozmnożenie" tego samego tekstu na dwa (lub więcej) okna. Zmieniając tekst w dowolnym z takich okienek powodujemy jego zmianę we wszystkich oknach związanych z tym tekstem. Jest to idealne, gdy pracujemy na długim tekście i chcemy go oglądać lub poprawiać w kilku odległych od siebie miejscach. Pracę z dużymi tekstami ułatwia również możliwość zdefiniowania aż dziesięciu "bookmarków" (CED może ich mieć tylko trzy) definiowanych wciśnięciem klawiszy Amiga+SHIFT razem z klawiszami od F1 do F10, a wywoływanych wciśnięciem klawisza SHIFT razem z jednym z klawiszy funkcyjnych. Ponadto możliwe jest ukrycie partii tekstu, których już nie chcemy zmieniać w celu szybszego poruszania się po tekście. Ukryte części tekstu nadal istnieją w dokumencie, ale nie są wyświetlane. Aby zdefiniować taki niewidoczny fragment tekstu, należy na jego początku wpisać kombinację znaków /*fs*/, a na końcu: /*fe*/. Następnie trzeba umieścić kursor wewnątrz ukrywanego bloku i wybrać funkcję "Folds/Hide" z menu "Tools". Aby ukryty blok był ponownie wyświetlany, należy ustawić kursor w tekście /*fs*/ (nie jest on ukrywany razem z blokiem) i wybrać funkcję "Folds/Show" lub funkcję "Folds/Show All", która powoduje wyświetlanie wszystkich ukrytych fragmentów w dokumencie. Należy jednak pamiętać, że przy drukowaniu wszystkie ukryte bloki zostaną zawsze wydrukowane. Wydrukowane zostaną również ciągi znaków definiujące początek i koniec niewidocznych fragmentów. Tak więc przed wydrukiem należy je usunąć z tekstu.

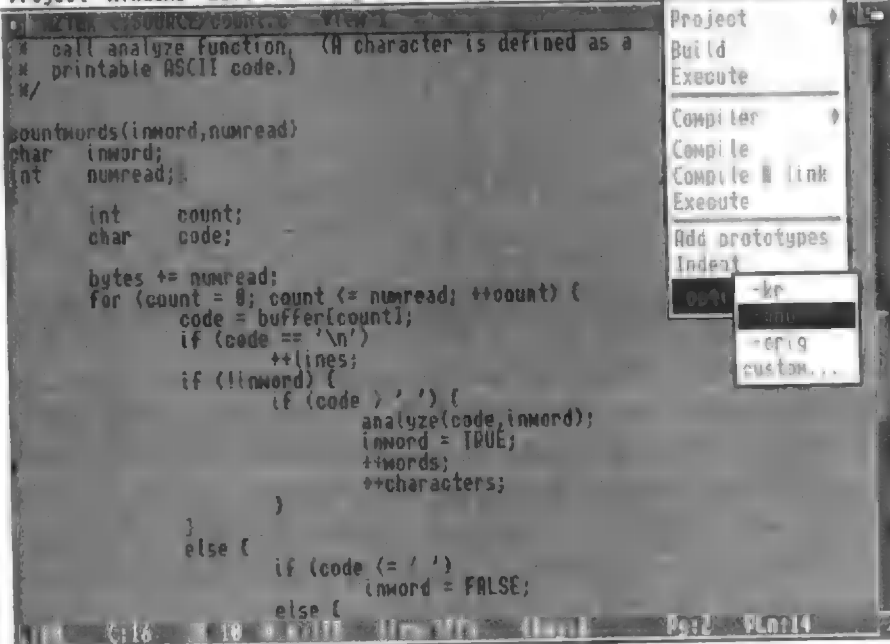
Przy wywoływaniu funkcji z klawiatury w przeważającej większości kombinacje klawiszy są identyczne jak

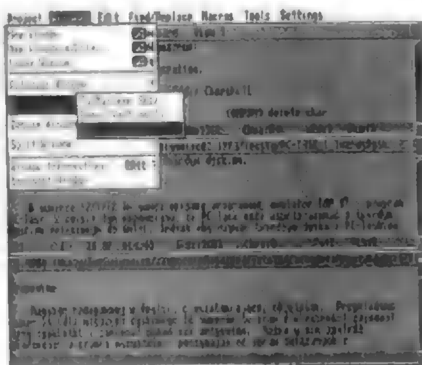




w edytorze CED, co ■ pewnością docenią ci, którzy wcześniej go używali. Lecz jeżeli coś nam nie odpowiada, możemy wykorzystać główną zaletę Edge, którą jest możliwość dowolnego skonfigurowania wszystkich menu i wszystkich kombinacji klawiszy. Możemy nawet zmienić teksty wszystkich komunikatów jakie wyświetla Edge. Przy odrobinie zacięcia możemy Edge całkowicie spolszczyć.

Zmiana parametrów Edge jest dziecinnie prosta, gdyż wszystko jest zapisane w plikach tekstowych, które można zmienić dowolnym programem (choćby przy pomocy Edge). Najciekawszymi plikami konfiguracyjnymi są "edge.menus" zawierający definicję menu edytora i "edge.teksts", w którym umieszczone są teksty komunikatów wyświetlanych przez program. W równie prosty sposób można zmienić przyporządkowanie klawiszy, a nawet reakcje na wciśnięcie przycisków myszy. Aby ułatwić edycję parametrów, edytor zawiera standardowo menu "Settings", w którym możemy prosto zmienić większość opcji tego edytora. Przykładowo wybierając z tego menu opcję "Local..." możemy zdefiniować (dla każdego okna z tekstem osobno) punkty tabulacji, lewy i prawy margines, włączyć lub wyłączyć automatyczne przenoszenie wyrazów itd. Natomiast wybierając "General..." mamy dostęp do ustawień dotyczących okienek, drukarki i ekranu, na którym Edge pracuje. Ekranem tym może być ekran Workbencha, ekran dowolnego programu używającego tak zwanego ekranu "public" lub własny ekran Edge. Ekran otwierany przez Edge może być w dowolnej z możliwych rozdzielczości do uzyskania na danego typu Amidze. Jeżeli jednak chcemy wykorzystywać tryby AGA, musimy mieć uruchomiony odpowiedni sterownik monitora (bez niego będziemy mogli jedynie użyć takiej rozdzielczości jaką ma w danej chwili ekran Workbencha). Jeśli mamy zwykły monitor (lub telewizor), wystarczy użyć sterownika o nazwie "PAL". Po jego uruchomieniu możemy nawet pracować w trybie Super Hires interlace (chociaż na zwykłym monitorze tekst jest raczej mało czytelny). W menu "General..." możemy również zadecydować, czy Edge ma używać requestera z biblioteki ASL (pole ASLReq włączone), czy też własnego - dosyć rozbudowanego - requestera (ASLReq wyłączone). Tam również, po kliknięciu w pole "Printer", mamy możliwość ustawienia parametrów drukowania. Możemy mianowicie zdefiniować wygląd nagłówków oraz stopek, długość strony i szerokość marginesów.





Edge posiada szereg ułatwień dla programistów. Tak samo jak EdWord ma funkcję sprawdzania poprawności liczby otwartych i zamkniętych nawiasów (menu "Tools/Move to/Matching brackets"). Oprócz tego na dysku z Edge znajduje się plik "Ed ge.compiler.menu", który po dołączeniu do zbioru "Edge.menus" dodaje do menu na górnej belce pozycję "Compiler". Zawarte w nim funkcje umożliwiają bardzo wygodną współpracę edytora Edge z kompilatorem SAS/C. Wzorując się na tym przykładzie można samodzielnie przystosować Edge do współpracy z dowolnym kompilatorem.

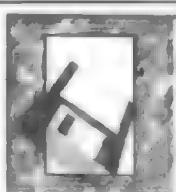
W liście 114 rozkazów ARexxa, w jakie wyposażony jest Edge, można znaleźć rozkaz porównywania słowa ze słownikiem. Jednocześnie dołączony jest przykładowy słownik słów kluczowych języka C. Możliwe jest więc używanie Edge do sprawdzania poprawności wpisanych tekstów. Niestety jednak do Edge nie dodano skryptu w ARexxie, który wykonywałby tę czynność (chociaż istnieje skrypt dodający wskazane słowo do słownika).

Przy omawianiu Edge nie można zapomnieć o bardzo dopracowanej funkcji Undo. Jest ona tak rozbudowana, że możemy skasować całkiem spory tekst literka po literce, a potem go odtworzyć (również litera po literze) za pomocą Undo. Tak dobre Undo posiada jedynie CED.

Wspomnę jeszcze, że na dysku z Edge znajdują się wersje tego programu na procesor 68000 oraz na procesor 68020 i wyższe (chociaż wersja na 68000 działa poprawnie również na wyższych procesorach).

* * *

Podsumowując, można stwierdzić, że EdWord jest naprawdę niezłym narzędziem dla osób piszących programy. Jednak Ci, którzy używali wcześniej CEDa, mogą mieć problemy ■ nauczeniem się dosyć dziwacznego obłożenia klawiatury. Edge natomiast jest doskonałym edytorem tekstu dla wszystkich, nie ustępującym pod żadnym względem programowi CED 3.5, ■ czasami nawet go przewyższającym. Jedyną wadą Edge (o ile jest to wada) jest praca tylko pod systemem 2.0 i wyższymi, ale przecież czas systemu 1.3 dobiega końca. ❑



VOYAGER

Tomasz [illegible]

Coraz rzadziej spoglądamy na usiane gwiazdami niebo. Przytłoczeni codziennością i oślepieni miejskim światłem nie dostrzegamy jego dostojnego piękna. Jeżeli zdarzy się nam podnieść wzrok do góry, wówczas przychodzi nam do głowy szereg znanych nazw: Wielka Niedźwiedzica, Andromeda, Krzyż Południa, itp. Gdzie znaleźć na niebie te gwiazdozbiory?

Biorąc pierwszy raz do ręki atlas nieba nie sposób się w nim połapać. Dziwne mapki pełne kropek i niezrozumiałych znaków. Jak odnieść to do rzeczywistości nocnego nieba? Bez obaw, nasza "przyjaciółka" nam w tym pomoże.

"Sky", "Distans Sun", "Galileo", "Voyager" oraz cały szereg programów PD powstało właśnie po to, aby ułatwić życie początkującym astronomom. Po zapoznaniu się z niektórymi z nich, ostatecznie zdecydowałem się na dłuższą zabawę z "Voyagerem". Przyczyn takiej decyzji jest kilka. Przede wszystkim program ten zawiera w sobie zdecydowanie najwięcej informacji o wszelkiego typu obiektach kosmicznych, widocznych na niebie. Obsługa programu, pomimo olbrzymiej ilości funkcji, jest niezwykle prosta. Jednak najważniejszą jego zaletą jest pełna, dynamiczna symulacja nieba. Pod tym pojęciem kryje się możliwość oglądania nieba z praktycznie dowolnego miejsca na ziemi i do tego w ruchu! Pozwala to na dokładne prześledzenie wszystkich zmian widoku nieba w zależności od pory roku i określonej godziny. Program pozwala nam np. zobaczyć jak wyglądało niebo w chwili narodzin Jezusa (rok pierwszy, chociaż nie wszyscy mogą się z tym zgodzić) lub jak powinno wyglądać powiedzmy w roku 2001. Widzimy więc, że komputer pozwala nam na znacznie więcej niż jakakolwiek książka. Jako ciekawostkę można tu podać fakt, że tego typu programy (oczywiście w zupełnie innej skali) używane są do całkiem poważnych badań naukowych. Symuluje się np. rozwój galaktyk, na przestrzeni setek milionów i miliardów lat, w zależności od parametrów początkowych. I co dziwne często okazuje się, że wyniki pracy komputera zaskakują nawet samych twórców takiego specjalistycznego oprogramowania.

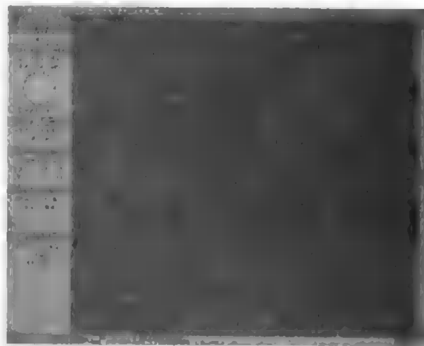
Przyjrzyjmy się teraz bezpośrednio "Voyagerowi". Standardowo program ten zajmuje aż dziewięć dysków. Do prawidłowej pracy wystarcza tylko jeden z nich zawierający główny program. Na pozostałych dyskach znajduje się całe mnóstwo zdigitalizowanych zdjęć przedstawiających najciekawsze i najbardziej znane obiekty kosmiczne. Dopiero zainstalowanie na dysku twardym pozwala wykorzystać pełną wersję programu.

"Voyager" uruchamia się na wszystkich modelach Amig posiadających 1 MB pamięci RAM. W przypadku modeli A500 i A600 niektóre operacje będą się odbywały dość ślamazarnie, ale w granicach przyzwoitości. Zdarzyło mi się podczas "radosnego poznawania" programu zająć go na pełnych piętnaście minut. Po zakończeniu obliczeń okazało się, że kazałem programowi obliczyć położenie pewnej planetoidy w roku 2000. Cóż, na błędach się uczymy. Inną cenną zaletą "Voyagera" jest to, iż uruchamia się on w trybie pracy Workbencha, jako dodatkowy proces w tle.

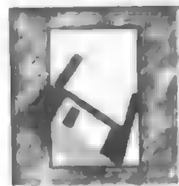
Przybliżając czytelnikom "Podróżnika" (tak można bowiem przetłumaczyć angielską nazwę "Voyager") będę starał się przedstawiać poszczególne funkcje w powiązanych ze sobą blokach tematycznych. W wielu miejscach stanie się koniecznością wprowadzenie dodatkowych pojęć związanych bezpośrednio z astronomią. Bez nich bowiem cały zasób informacji zawartych w programie będzie niejasny i nie w pełni wykorzystany.

Podczas uruchamiania programu pojawia się okno zawierające informacje o autorach programu, wolnej ilości pamięci Chip i Fast RAM oraz wielkości ekranu, na którym "Voyager" będzie pracował. Do tych informacji można również dostać się poprzez opcję "File" - "About Voyager".

Ekran roboczy wygląda tak, jak przedstawiono to na rysunku 1. Jego centralną część zajmuje oczywiście pewien wycinek nieba. U góry tradycyjnie rozgościło się rozwijane menu, natomiast lewa strona została zarezerwo



Rysunek 1.



wana dla panelu dublującego podstawowe opcje programu związane ze sterowaniem wspomnianej wcześniej, dynamicznej symulacji nieba. Włączanie i wyłączanie panelu odbywa się za pomocą "Sky" - "Show sky panel". Wyłączenie tej opcji powiększa widziany fragment nieba.

Zajmijmy się najpierw wspomnianym panelem. Najwazniejszą z opcji tam znajdujących się jest "Location". Najedźmy więc na nią kursorem i wciśnijmy lewy przycisk myszy. Pojawi się wówczas okno zamieszczone na rysunku 2. Dokonujemy tutaj wyboru miejsca na Ziemi, z którego chcemy obserwować niebo. Autorzy programu starali się zbliżyć maksymalnie do rzeczywistości, stąd też posługujemy się w tym celu odpowiednio przedstawionym globusem. Chcąc go przesunąć wystarczy w wybranym miejscu dwukrotnie kliknąć lewym przyciskiem myszy. Same zarysy kontynentów niewiele nam dadzą, dlatego posłużymy się dodatkowymi możliwościami w postaci opcji "Show Grid" i "Show Cities". Pierwsza z nich zaznacza na globusie siatkę złożoną z południków i równoleżników, druga natomiast przedstawia w postaci żółtych punktów położenie większości dużych miast. Z Polski niestety zakwalifikowała się jedynie Warszawa.

Chcąc łatwo znaleźć interesującą nas miejscowość można powiększyć centralny fragment widocznej części globusa, posługując się opcjami "Mag 2*" lub "Mag 4*". Po zastosowaniu tej ostatniej, na mapce dodatkowo pojawią się nazwy uwidocznionych miast (rysunek 3).

Ustawmy zatem globus na widok Europy i powiększmy go czterokrotnie. Wskażmy teraz (jednokrotne przyciśnięcie lewego klawisza myszy) naszą stolicę ("Warsaw"). Powinny się wówczas pojawić następujące informacje:



Rysunek 2.



Rysunek 3.

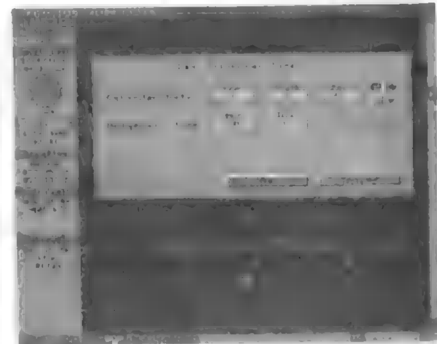
- długość geograficzna wschodnia ("Longitude E") 21° 0',
- szerokość geograficzna północna ("Latitude N") 52° 15',
- strefa czasu: 1.0 (czyli strefa, w której przyjmuje się godzinę różnicy w stosunku do czasu Greenwich).

Skoro mamy już odpowiednie "miejsce akcji" pora więc na ustalenie jej czasu. Oficjalnie przyjęto dwa sposoby podawania czasu: lokalny i uniwersalny. Pierwszy z nich nazywany również czasem strefowym, to rodzaj odmierzania czasu, którym posługujemy się na codzień. Wynika on z podziału kuli ziemskiej na strefy czasowe (dokładnie rzecz biorąc podział ten dokonywany jest odpowiednimi południkami, rozmieszczonymi co 15°). Każda taka strefa decyduje o przesunięciu o 1 godzinę wskazówek zegara w stosunku do czasu uniwersalnego. Ten ostatni odpowiada natomiast średniemu czasowi słonecznemu (średni czas słoneczny związany jest z ruchem słońca po nieboskłonie) południka przechodzącego przez Greenwich. Ogólnie objawia się to tym, że jeżeli np. w Warszawie będzie godzina 9:47, wówczas w tej samej chwili w angielskiej miejscowości Greenwich odczytamy godzinę 8:47, a w Dallas 2:47 (w nocy oczywiście).

Odpowiednie wartości dotyczące czasu uniwersalnego ustawiamy w menu "Univ Time" (rysunek 4). Wpisujemy tam:



Rysunek 5.



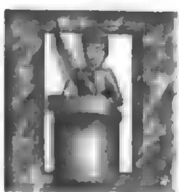
Rysunek 4.

- "Calendar date" - data kalendarzowa ("Year" - rok, "Month" - miesiąc, "Day" - dzień). Przyciskami "AD" i "BC" ustalamy odpowiednio, czy wpisana data odnosi się do ery nowożytnej (po narodzinach Chrystusa), czy też dotyczy starożytności. Jeżeli wpiszemy rok jako wartość ujemną, wówczas program automatycznie przeskoczy pomiędzy "AD" a "BC". Jako "Universal Time" podajemy interesującą nas godzinę ("Hour") wraz z minutami ("Min"). Podobnie postępujemy przy ustalaniu czasu lokalnego "Local Time". Jedyna różnica dotyczy tego, iż przy podawaniu godziny program posługuje się zegarem 12 godzinnym i należy go poinformować, czy podana godzina odnosi się do czasu od północy do południa ("PM") czy od południa do północy ("AM"). Należy pamiętać, że oba sposoby odmierzania czasu są ze sobą wzajemnie powiązane. Jakakolwiek zmiana jednego z ustawień, automatycznie pociąga za sobą zmianę drugiego ustawienia.

W ten oto sposób przygotowaliśmy program do rozpoczęcia właściwej pracy. Mam nadzieję, że coraz cieplejsze noce i zbliżające się lato zmobilizują Czytelników do spędzenia kilku godzin pomiędzy ekranem komputera, a rozgwieżdżonym niebem. Zainteresowanych możliwościami "Voyagera" zapraszam na spotkanie w kolejnym numerze "Amigowca".

Na marginesie, od autora:

Mam prośbę do potencjalnych Czytelników moich artykułów. Otóż nie wiem jak mocno powinienem zagłębić się w wyjaśnianie astronomicznych pojęć pojawiających się w tym programie. Z jednej strony są one niezbędne dla pełnego wykorzystania możliwości "Voyagera", z drugiej nie chciałbym zanudzać Państwa wywodami szeroko już opisanymi w stosownej literaturze. Dlatego byłbym wdzięczny za przesłanie na adres redakcji uwag dotyczących tego tematu. ❑



"Dlaczego IBM amigowacieje i nikt nie zwraca na to uwagi?"

W numerze 2/94 Amigowca niezastąpiony Tomek Kokoszyński przedstawił artykuł zatytułowany Z życia komputerowego psa .... Odniosłem wrażenie, że został on napisany ku pokrzepieniu serc Amigantów. I słusznie, bo co mniej odporni z nich mogą ulec powszechnemu zalewowi pecetczyzny. Chciałbym dorzucić i swoją cegiełkę na ten chwalebny cel.

Zanim przejdę do zasadniczej części artykułu, poświęcę trochę miejsca na:

"Gorzkie żale"

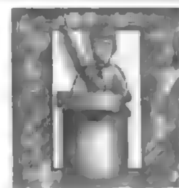
Jestem człowiekiem niezwykle opanowanym. Końcową fazą mojego zdenerwowania jest śmiertelny spokój. Dlatego przez długi okres czasu nie reagowałem na bzdury pojawiające się w czasopismach o tworach PC-podobnych, [redacted] dotyczące naszej poczciwej Amigi (i nie tylko). Niedawno np. dowiedziałem się, że maksymalnymi możliwościami graficznymi A4000 jest zastosowanie 4096 kolorów przy rozdzielczości 640 [redacted] 512 punktów (?). Tego typu techniczne nieścisłości można [redacted] jeszcze "przeboleć", naprawdę denerwujące jest "poniżanie przeciwnika". Na skraj wybuchu doprowadziły mnie dwie rzeczy, którymi uraczyło nas pewne wydawnictwo spod znaku "wilczka". Pierwszą z nich był rysunek przedstawiający dwóch ludzi, z których jeden obarczony transparentem z napisem Apple klęczy, całując sztandar IBM-a. Fuj! Jeżeli kiedykolwiek zdarzenie takie będzie miało miejsce, wówczas role będą raczej odwrotne. A tak na marginesie to wydaje mi się, że znak firmowy i nazwa Apple jest zastrzeżona. Nie czepiajmy się szczegółów. Druga rzecz jest znacznie bardziej smutna i dotyczy następującego podziału zastosowań mikrokomputerów, przedstawionego [redacted] łamach tego samego czasopisma:

- IBM - biura, DTP, edukacja,
- Atari - DTP, muzyka,
- Amiga - rozrywka.

Również w szeregu testów i porównań dotyczących oprogramowania oraz sprzętu, możliwości Amigi są skrzętnie pomijane. Niewiedza to, czy zwykła złośliwość. Trudno powiedzieć. Jedno jest pewne, takie zachowanie oddaje firmie Commodore przysłowiową "niedźwiedzią przysługę". Dziwi mnie tylko fakt, że polskie przedstawicielstwo tej firmy (o którego naturalnym zgonie "Amigowiec" informował niedawno [redacted] szczerym brakiem jakiegokolwiek żalu) nie reagowało na tego typu kryptoreklamę. Właściwie można powiedzieć, że owo przedstawicielstwo w ogóle niczego nie robiło, poza samozadowoleniem z masowej sprzedaży najprostszych modeli A500, A600 i A500+. A przecież typoszereg Amig jest znacz nie bogatszy: A1200, A2000, A3000, A3000T, A3000UX, A4000/030, A4000/040, czy też już nieco zapomniane A1000 i A2500. Tyle też i nieco spóźnionych żalów.

"Dlaczego IBM amigowacieje", czyli wstęp do obalenia niecnych insynuacji o nieprofesjonalności Amigi.

Na początku był chaos. Wynalezienie tranzystora, [redacted] następnie układu scalonego stało się motorem rozwoju elektroniki. Powstało szereg mniej lub bardziej udanych konstrukcji, będących praprzodkami współczesnych procesorów. Tworzyły się i upadały różnego typu firmy i przedsiębiorstwa. Dopiero gdzieś na początku lat siedemdziesiątych [redacted] tego tygla wyłoniły się dwa kolosy do dziś wiodące prym w dziedzinie specjalizowanych układów scalonych. Są to oczywiście firmy Motorola i Intel. Z upływem czasu okazało się, że poszły one w różnych kierunkach rozwojowych. Motorola postawiła na optymalizację procesorów pod kątem skrócenia listy rozkazów i [redacted] ich przetwarzania, natomiast Intel nadrabiał wszystkie braki zwiększaniem szybkości tzw. zegara (16, 33, 50, 66 MHz itp.). Tak, jak i różne były procesory, tak i różne stały się komputery, w których je zastosowano. Intelowska seria 80X86 znalazła się w komputerach cyklu PC/AT, natomiast serią 680XX Motoroli zainteresowały się aż trzy firmy: Apple, Commodore i Atari. Jako pierwszy pojawił się Apple'owski Macintosh. Następna była już nasza "przyjaciółka" w postaci modelu A1000 z procesorem 68000. Przypomnę tylko, że był to rok 1985, w którym "królem" mikrokomputerów był IBM oparty na procesorze 80286. Wyobraźmy sobie kontrast pomiędzy użytkownikiem tego "cuda" mozolnie wklepującym instrukcje MS-DOSa, [redacted] posiadaczem Amigi wygodnie rozpartym [redacted] fotelu i przesuwającym kursor na ekranie za pomocą myszy. Pomimo pojawienia się Windowsa z całym szeregiem różnego typu aplikacji, nadal pierwsze kontakty z Amigą moich IBMowo zorientowanych kolegów kończą się próbą samobójczego skoku przez okno. Dlaczego tak się dzieje? Otóż nowe środowisko pracy i odpowiadające mu oprogramowanie postawiło odpowiednio wygórowane wymagania. Okazało



się, że standardowa konfiguracja typu: jeden procesor + zwykła karta grafiki + złącza, nie jest w stanie udźwignąć nowo narzuconych obowiązków. Sytuację tę najlepiej oddaje anegdotka, krążąca ostatnio wśród komputerowej braci, o następującej treści: "Co się dzieje kiedy chcesz odpalić nowy wspaniały program na Amidze? Pojawia się wówczas komunikat: dołóż RAMu, dołóż RAMu. Analogiczna próba na IBMie kończy się innym napisem: wymień płytę główną, wymień płytę główną".

Oczywiście można temu zaradzić. Wystarczy kupić komputer klasy 386DX lub 486 z Local Busem, procesorową kartą grafiki, kartą z układami dźwiękowymi, itd. Skąd my to znamy? Przecież od dawna posługujemy się takim zestawem. Każda Amiga ma w standardowym wyposażeniu cały zestaw dodatkowych procesorów i specjalnych układów scalonych. Ponieważ komputer nasz był od podstaw pomyślany jako wieloprocesorowy, stąd posiada on zasadniczą przewagę nad konkurencją. Objawia się to kilkoma podstawowymi elementami. Po pierwsze programy napisane na Amigę zajmują mniejszą ilość pamięci od swoich odpowiedników na IBMie. Chcąc np. wyświetlić coś na ekranie nie musimy badać konfiguracji systemu i pisać specjalnego drivera dla karty graficznej, lecz wystarczy wydać odpowiedni rozkaz procesorowi graficznemu. Tak samo ma się sprawa z dźwiękiem i innymi operacjami wykonywanymi przez komputer. Ta sytuacja owocuje jeszcze jedną zaletą. Otóż na zbliżonych cenowo komputerach (szczerze mówiąc trudno tu stosować inne kryterium porównawcze) analogiczne programy będą działały szybciej.

W tym momencie ktoś może zapytać, dlaczego wszyscy gromadnie nie przesiadają się na Amigi? Powodów jest kilka. Po pierwsze działa tu element psychologiczny. Jeżeli ktoś pracuje w biurze, banku lub innej tego typu instytucji i korzysta tam z komputera IBM, wówczas uważa, że lepiej mieć w domu taki [redacted] komputer jak w pracy. Wydaje się, iż jest w tym trochę racji, a później zaczynają się kłopoty. Bowiem przydałby się trochę lepszy dźwięk, więcej koloru, no i te animacje mogłyby się mniej "haczyć". Drugim powodem jest PROFESJONALIZM Amigi. Komputer ten został poświęcony od A do Z jednemu tylko celowi - grafice. Począwszy od standardowo rozbudowanych możliwości graficznych, poprzez oprogramowanie, aż po fakt dobrania częstotliwości pracy procesora jako krotność częstotliwości z jaką tworzony jest obraz telewizyjny. Praktycznie rzecz biorąc większość ludzi zajmujących się tworzeniem animacji komputerowych stawiała swoje pierwsze kroki na Amigach. W chwili obecnej jest to bowiem jedyny komputer wypełniający lukę pomiędzy super profesjonalnymi stacjami roboczymi, a mikrokomputerami osobistymi. Przy odrobinie cierpliwości można nawet na A600 bez dysku twardego przygotować i nagrać na kasetę prostą czołówkę video.

Wystarczy tych zachwytów, dobry produkt sam się obroni, trzeba dać mu tylko szansę. A właśnie, puśćmy wodze wyobraźni. Co by było, gdyby...

Gdyby firma Commodore od początku bardziej zaangażowała się w promowanie Amigi. Gdyby zamiast konkurować z Atari ST podjęto próbę stworzenia wspólnego systemu operacyjnego, opartego na procesorze 680XX, pozwalającego na powtórzenie sukcesu Apple'a w USA (czyli wyparcia IBMa [redacted] większości zastosowań biurowych i akademickich). Gdyby w latach dziewięćdziesiątych polskie przedstawicielstwo firmy Commodore zajęło się promowaniem Amigi wśród nowo powstających agencji reklamowych i telewizji regionalych. Gdyby, gdyby... Gdyby to wszystko się udało, wówczas "Amigowiec" miałby 160 stron, pełen kolor i przynajmniej 100 tys. czytelników. A tak mamy, to co mamy.

Wiadomość z ostatniej chwili.

Prawdopodobnie już wkrótce nie będzie się w ogóle o co spierać. Koncerny: Apple, IBM i Motorola doszły do wniosku, że dalsza wspólna rywalizacja do niczego nie zaprowadzi i opracowały wspólny model procesora RISC-Power PC 6XX. Jego odpowiednia zabudowa ma pozwolić na równoczesne korzystanie z aplikacji pracujących pod systemami Apple System 7, Windows NT i IBM OS/2. W dalszej kolejności do "rodzinnego grona" mają dołączyć stacje robocze SUN oraz IBMowskie RS/6000. Mogłoby się wydawać, że dla nas takie nowinki [redacted] odległą melodią przyszłości, gdyby nie pewien drobny fakt. Oto któregoś ranka w "śniadaniowej telewizji polskiej", w ramach komputerowych ciekawostek, pojawił się pewien uśmiechnięty pan architekt. Stwierdził on, że praca na Macintoshu zaopatrzonym w Power PC ulega kilkukrotnemu przyspieszeniu oraz, że system taki otwiera przed użytkownikiem zupełnie nowe możliwości. Na potwierdzenie tych słów pokazano ekran systemu Apple'a, na którym otwarto Windowsa, uruchomiono "Paint Brusha", wykonano rysunek i bezpośrednio przeniesiono go do oryginalnego Macintoshowego programu! Skoro sprawy zabrnęły już tak daleko, to pozostaje mieć nadzieję, że firma Commodore nie pozostanie z tyłu. Od dawna krążyły bowiem informacje o nowym modelu Amigi opartym na procesorze 68060 i mającym możliwość pracy wielosystemowej (Workbench + Windows NT). W aktualnie powstałej sytuacji większe nadzieje wiązałbym z zastosowaniem wspomnianego wcześniej procesora Power PC. Szereg znaków na Niebie i Ziemi sugeruje, że może tak być w istocie. Połączenie w jedno dotychczasowych możliwości sprzętowych i programowych Amigi z szybkością działania RISCa, powinno zaowocować zupełnie nowym jakościowo komputerem. Tylko o co wówczas będziemy się kłócili z IBMowcami? ❑

KUPON KONKURSOWY

nr 4/94

m24	k11					
n19	e2	a15	[illegible]	[illegible]	d3	t25
c26	k18	u5	g2	t2[illegible]	e19	w19
i7	p27	a22				
r6	z8	e23	w12	n2[illegible]		
n4	h13	[illegible]	b13	z2[illegible]	d[illegible]	s11

Imię Nazwisko: ....

Adres: ....

Wiek(nie trzeba wypełniać): ....

Zawód(nie trzeba wypełniać) ....

KURS ASEMBLERA cz. 8

Zgodnie z obietnicą sprzed miesiąca, dokończymy dzisiaj pisania programu prostego kalkulatora, który - mam nadzieję - da Wam ogólne wyobrażenie o pisaniu programów (na początek bardzo prostych) w języku assemblera.

Jak pewnie pamiętacie, nasz kalkulator ma na razie tylko klawisze służące do wprowadzania cyfr. Na początek dodamy więc do niego klawisze działań matematycznych: '/', '*', '-', '+', klawisz wyniku '=' i klawisz kasowania 'AC'. W tym celu stworzymy sześć nowych gadżetów: "KasujGAD", "DzielenieGAD", "MnozenieGAD", "OdejmowanieGAD", "DodawanieGAD" i "WynikGAD" (patrz listing). Tak, jak każdy gadżet wyboru cyfr miał przypisany w polu gg_gadgetID swój własny numer służący do określenia, który gadżet został wciśnięty, tak i nowe gadżety mają swoje numery:

10 - klawisz kasowania "AC"
11 - klawisz dzielenia "/"
12 - klawisz mnożenia "*"
13 - klawisz odejmowania "-"
14 - klawisz dodawania "+"
15 - klawisz wyniku "="

Wartości te są wybrane w całkowicie dowolny sposób, lecz musimy pamiętać, która wartość odpowiada jakiemu klawiszowi, gdyż tylko na jej podstawie będziemy wiedzieć jakie działanie mamy wykonać.

Tym, którzy mają już wpisany listing z poprzedniego odcinka chcę zwrócić uwagę, że po wpisaniu nowych fragmentów z definicjami gadżetów operacji matematycznych, należy jeszcze zmienić w gadżecie "Klawisz9GAD" pole gg_NextGadget (to jest ta wartość w linii z etykietą) z zera na adres kolejnego gadżetu, czyli "KasujGAD".

Kalkulator z wszystkimi klawiszami wygląda następująco:



Gdy już mamy zaprogramowane wszystkie gadżety, możemy się zająć procedurami obliczeniowymi.

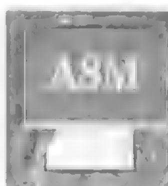
Wprowadzane przez użytkownika liczby otrzymujemy w kodzie ASCII (w kodzie ASCII każdej cyfrze (i literze) odpowiada jedna wartość), który musimy przeliczyć na zwykłą liczbę, na której będziemy mogli przeprowadzać operacje matematyczne. Jak powinna działać procedura zamieniająca ciąg znaków ASCII na liczbę? Najprostszy sposób to pomnożenie ostatniego (pierwszego z prawej) znaku przez 1, przedostatniego przez 10, kolejnego przez 100, itd., a następnie zsumowanie wszystkich składników. Najbardziej prymitywna procedura mogłaby wyglądać w następujący sposób:

```
ASCII_Liczba1   lea     Bufor+5,a0 ; adres konca ciagu ASCII
                moveq   #0,d0      ; tu będzie wynik
                moveq   #0,d1
                move.b  -(a0),d1   ; ostatni znak ASCII
                subi.b  #48,d1
                mulu    #1,d1      ; mnożenie przez 1
                add.l   d1,d0      ; sumowanie składników
                moveq   #0,d1
                move.b  -(a0),d1   ; przedostatni znak ASCII
                subi.b  #48,d1
                mulu    #10,d1     ; mnożenie przez 10
                add.l   d1,d0      ; sumowanie
                moveq   #0,d1
                move.b  -(a0),d1   ; trzeci znak od konca
                subi.b  #48,d1
                mulu    #100,d1    ; mnożenie przez 100
                add.l   d1,d0      ; sumowanie
                moveq   #0,d1
                move.b  -(a0),d1   ; czwarty znak od konca
                subi.b  #48,d1
                mulu    #1000,d1   ; mnozenie przez 1000
                add.l   d1,d0      ; sumowanie
                moveq   #0,d1
                move.b  -(a0),d1   ; piąty znak od konca
                subi.b  #48,d1
                mulu    #10000,d1  ; mnożenie przez 10000
                add.l   d1,d0      ; sumowanie
                rts                ; w d0 jest wynik
Bufor           dc.b    '12345'
```

Wyjaśnienia wymaga wpisywanie 0 do rejestru D1 przed każdym pobraniem znaku z bufora (instrukcje moveq #0,d1 przed move.b -(a0),d1). Otóż instrukcja move.b -(a0),d1 zmienia tylko najmłodszy bajt rejestru D1, a wykonywana później instrukcja mnożenia mulu #nn,d1 mnoży całe młodsze słowo (dwa bajty) rejestru D1 przez liczbę, więc gdybyśmy nie wyzerowali pozostałych bitów D1, to wynik mnożenia byłby błędny. Jednocześnie po wykonaniu mnożenia w D1 znajduje się wynik mnożenia, więc kasowanie rejestru musimy za każdym razem powtarzać.

Niezrozumiałe może być także odejmowanie liczby 48 od zawartości pobranej z bufora (instrukcje subi.b #48,d1). Wartość 48 to liczba odpowiadająca znakowi "0", tak więc odjęcie kodu "0" od kodu "0" daje nam 0, odjęcie kodu "0" od kodu "1" daje nam 1, itd. W rezultacie otrzymujemy rzeczywistą wartość danej cyfry. (Większość assemblerów przyjmuje zamiast #48 zapis #'0', który w tym przypadku jest bardziej czytelny.) Powyższą procedurę można oczywiście zrealizować z użyciem pętli:

```
ASCII_Liczba2   lea     Bufor+5,a0         ; adres konca ciagu ASCII
                lea     TabelaMnozen,a1    ; tabela z mnożnikami
```




```
         moveq   #0,d0     ; tu bedzie znajdowal się wynik
ASCII_L1 moveq   #0,d1
         move.b  -(a0),d1  ; pobranie jednej cyfry ASCII
         subi.b  #'0',d1
         mulu    (a1)+,d1  ; pomnożenie przez liczbę z tabeli
                           ; (1,10,100,1000,...)
         add.l   d1,d0     ; dodanie składnika
         cmpa.l  #Bufor,a0 ; sprawdzenie, czy już wszystkie cyfry
         bne     ASCII_L1  ; jeśli nie, to skok do "ASCII_L1"
         rts
TabelaMnozen: dc.w 1
         dc.w    10
         dc.w    100
         dc.w    1000
         dc.w    10000
Bufor:   dc.b    '12345'
```

Pokazana tutaj metoda ma pewną wadę. Polega ona na tym, że przy jej wykorzystaniu możemy przeliczać jedynie liczby zapisane na nie więcej, niż pięciu znakach. Dzieje się tak dlatego, że gdyby liczba była zapisana - przykładowo - na sześciu znakach, to pierwszą cyfrę z lewej musielibyśmy mnożyć przez 100000. A 100000 nie da się przecież zapisać w jednym słowie, więc nie można stosować instrukcji mulu, która mnoży ze sobą tylko liczby w rozmiarze słowa (a więc od 0 do 65535). Zamiast mulu trzeba by więc zastosować procedurę mnożącą liczbę 16-bitową przez liczbę 32-bitową lub zamiast mnożenia stosować wielokrotne dodawanie (Np. mnozenie 10*20 to to samo, co dziesięciokrotne dodawanie 20). Metoda dodawania jest bardziej czasochłonna, jednak w naszym przypadku będziemy musieli wykonać maksymalnie 9 dodawań na cyfrę, więc nie będzie to zajmowało zbyt dużo czasu. Procedura wykorzystująca metodę wielokrotnego dodawania moze wyglądać następująco:

```
ASCII_Liczba3: lea   Bufor+8,a0
         lea     TabelaMnozen(pc),a1
         moveq   #0,d1      ; tu będzie przeliczana liczba
ASCII1:  cmpa.l  #Bufor,a0  ; sprawdzenie czy już ostania cyfra
         beq     ASCII4
         moveq   #0,d0      ; wyzerowanie rejestru D0
         move.b  -(a0),d0   ; pobranie jednej cyfry
         sub.b   #'0',d0    ; odjęcie kodu ASCII
         beq     ASCII3
ASCII2:  add.l   (a1),d1    ; tutaj jest realizowane
         subq.w  #1,d0      ; mnożenie poprzez
         bne     ASCII2     ; wielokrotne dodawanie
ASCII3:  addq.l  #4,a1      ; następna pozycja w tabeli
         bra     ASCII1
ASCII4:  rts                ; w D1 mamy już właściwą liczbę

TabelaMnozen: dc.l 1
         dc.l    10
         dc.l    100
         dc.l    1000
         dc.l    10000
         dc.l    100000
         dc.l    1000000
         dc.l    10000000
Bufor:   dc.b    '12345678'
```

Identyczna procedura została zastosowana w programie kalkulatora. W listingu ma ona nazwę "ASCII_do_liczby".

Po przeliczeniu ciągów ASCII na normalne liczby możemy zabrać się do napisania procedur wykonujących operacje matematyczne na tych liczbach.

Przyjmijmy, że nasz kalkulator będzie wykonywał działania jedynie na liczbach naturalnych, czyli liczbach całkowitych i większych lub równych zero. Zakres przyjmowanych liczb ograniczymy od góry największą liczbą możliwą do wyświetlenia na ośmiu cyfrach (tyle pozycji będzie miał nasz kalkulator), czyli poprawne będą liczby w zakresie od 0 do 99999999. Wszystkie spoza tego zakresu będą uważane za błędne i powodować będą wyświetlenie na wyświetlaczu napisu ERROR (ang. ERROR - błąd).

Na pierwszy ogień weźmy procedurę dodającą:

```
DodawanieProc: add.l d1,d0
         cmpi.l  #99999999,d0 ; większe liczby nie mieszczą
                              ; się na wyświetlaczu
         bcc     Blad         ; liczba zbyt duża
         ...
```

Jak widać nie ma w niej nic trudnego. Równie prosta jest procedura realizująca odejmowanie:

```
OdejmowanieProc: sub.l d1,d0
         bmi     Blad      ; liczba ujemna
         ...
```

Łatwa (wprost niesamowicie łatwa) do zrozumienia jest także procedura mnożąca:

```
MnozenieProc: mulu d1,d0
         ...
```

Trochę więcej problemów może stworzyć procedura dzieląca:

```
DzielenieProc: tst.l d1
         beq     Blad      ; dzielenie przez zero
         divu    d1,d0
         bvs     Blad      ; błąd występujący przy dzieleniu
                           ; bardzo dużej liczby przez bardzo małą
         andi.l  #$0000FFFF,d0 ; odcięcie reszty z dzielenia
                           ; (po DIVU znajduje się ona
                           ; w starszym słowie wyniku)
         ...
```

Na początku procedury realizującej dzielenie sprawdzamy zawartość rejestru D1, czyli dzielnika. Nie może on być równy zeru, gdyż dzielenie przez zero jest niewykonalne, a w przypadku procesorów 680xx wywołuje stan wyjątkowy, kończący się na Amigach przez Guru. Skok (bvs Blad) po instrukcji divu służy do wykrycia błędu dzielenia polegającego na dzieleniu bardzo dużej liczby przez bardzo małą. Może wówczas wystąpić taka sytuacja, gdy wynik dzielenia nie mieści się na pojedynczym słowie i instrukcja divu jest bezradna (dokładny opis rozkazów dzielenia znajduje się w Amigowcu 2/94). Nasz kalkulator nie będzie starał się poprawnie wykonać dzielenia (służy on przecież tylko celom dydaktycznym, a nie użytkowym) i zasygnalizuje błąd. Po wykonaniu dzielenia (rozkaz divu d1,d0) w rejestrze D0 znajduje się wynik dzielenia. Jednak należy pamiętać, że właściwy wynik znajduje się w młodszym słowie rejestru, natomiast w starszym słowie znajduje się reszta z dzielenia. Owa reszta nas nie interesuje, gdyż nasz kalkulator przeprowadza działania tylko na liczbach całkowitych. Aby pozbyć się reszty wykonujemy operację andi.l #$0000FFFF,d0, która spowoduje wyzerowanie najstarszych 16 bitów (czyli starszego słowa) rejestru D0.

Po wykonaniu działania musimy wynik z powrotem przekonwertować do znaków ASCII, celem wypisania go na wyświetlaczu naszego kalkulatorka. Procedura zamieniająca zawartość rejestru D0 na ciąg znaków ASCII wygląda w naszym programie następująco:

```
Liczba_do_ASCII: lea Bufor1,a0 ; adres bufora, gdzie
                               ; umieszczane będą znaki ASCII
         lea     TabelaDzielen(pc),a1 ; tabela liczb, przez które
                               ; będziemy dzielili liczbę
         tst.l   d0            ; czy konwertowana liczba
                               ; jest zerem
         beq     ToZERO        ; jeśli tak, to zajmiemy się nią osobno
ToASCII1: sub.l  (a1),d0       ; odjęcie od liczby wartosci z tabeli
         bcc     ToASCII2
         add.l   (a1)+,d0      ; jeśli wynik odejmowania był ujemny,
                               ; to musimy dodać to, co odjęliśmy i
         bra     ToASCII1      ; spróbować z mniejszą liczbą
                               ; z tabeli
ToASCII2: moveq  #0,d1
ToASCII3: addq.w #1,d1         ; jeśli wynik był >0, to odejmujemy
         sub.l   (a1),d0       ; wartość z tabeli i liczymy ile
         bcc     ToASCII3      ; razy się mieści w naszej liczbie
         add.l   (a1)+,d0      ; gdy już się nie mieści, to przechodzimy
                               ; na następną wartość w tabeli
         addi.b  #'0',d1       ; dodajemy "0", aby uzyskać
                               ; właściwy kod ASCII.
         move.b  d1,(a0)+      ; i wpisujemy go do bufora
         cmpa.l  #TabelaDzielenEnd,a1 ; sprawdzamy, czy już koniec
         beq     ToASCII6      ; jeśli tak, to skok do "ToASCII6"
ToASCII4: sub.l  (a1),d0       ; jeśli nie, to robimy to samo, co
         bcc     ToASCII2      ; na początku (pętla ToASCII1), lecz
         add.l   (a1)+,d0
         move.b  #'0',(a0)+    ; wstawiamy teraz do bufora zera
         cmpa.l  #TabelaDzielenEnd,a1 ; tu też sprawdzamy, czy już
         bne     ToASCII4      ; koniec wartości w tabeli
ToASCII6: rts
ToZERO:  move.b  #'0',(a0)+    ; gdy liczba do zamiany jest równa zero
```

```
                                ; to nie dokonujemy konwersji, lecz
            rts                 ; bezpośrednio wpisujemy "0"

TabelaDzielen.  dc.l    10000000,1000000,100000,10000,1000,100,10,1
Bufor1          dc.b    0,0,0,0,0,0,0,0
```

Działanie tej procedury polega na odejmowaniu od przeliczanej liczby coraz mniejszych potęg dziesiątki, poczynając od 10000000 (bo przy ośmiocyfrowej liczbie pierwsza z lewej cyfra odpowiada za ilość 10000000 w liczbie). Jeśli po odjęciu od badanej liczby wartości 10000000 uzyskamy przepełnienie, to będzie znaczyć, że 10000000 nie mieści się w naszej liczbie ani razu. W takim przypadku przechodzimy do niższej potęgi dziesiątki i odejmujemy liczbę 1000000 (oczywiście najpierw musimy dodać odjęte wcześniej 10000000, aby otrzymać pierwotną wartość liczby), gdy to nie da efektu, przechodzimy do 100000, potem do 10000, itd. Natomiast, jeśli któraś potęga dziesiątki mieści się w analizowanej liczbie n razy, to wartość tego n po dodaniu kodu ASCII dla '0', jest kodem ASCII pojedynczej cyfry. Po sprawdzeniu wejściowej liczby pod kątem zawartości wszystkich potęg dziesiątki otrzymujemy ciąg ASCII odpowiadający konwertowanej liczbie. W procedurze występują dosyć podobne do siebie pętle za etykietami "ToASCII1" i "ToASCII4". Pierwsza z nich jest wykonywana zanim napotka jakąś niezerową cyfrę i nie powoduje wstawienia do bufora nieznaczących zer (zera nieznaczące, to zera przed liczbą, które i tak nie zmieniają wartości liczby, np. w zapisie 0002550 trzy pierwsze zera są niepotrzebne, natomiast czwarte, ostatnie, jest istotne i nie można go usunąć). Natomiast druga pętla robi to samo co pierwsza, lecz wstawia zera, gdyż tym razem są one istotne.

W największym skrócie nasz kalkulator działa następująco: Po otwarciu okienka (procedury odpowiadające za otwarcie okna były już opisywane - Amigowiec 9-10/93) wywoływana jest procedura o nazwie "Kalkulator", która jest główną procedurą w naszym programie. W jej obrębie zajmujemy się zainicjowaniem wyświetlacza, czyli narysowaniem ramki dookoła niego i jego wyczyszczeniem (to znaczy wypisaniem na nim liczby "0"). Następnie zerujemy komórki pamięci, w których będziemy przechowywać nasze zmienne. Po tych czynnościach przygotowawczych możemy już zainteresować się, co robi użytkownik naszego programu. W tym celu wywołujemy procedurę "PobierzLiczbe", która czeka aż użytkownik wpisze liczbę i wciśnie klawisz operacji, którą chce wykonać. Od procedury otrzymujemy wówczas w rejestrze D0 - kod klawisza operacji, natomiast w rejestrze D1 - liczbę, którą wprowadził użytkownik. Po sprawdzeniu, czy w D0 nie ma wartości 255 (kod gadżetu zamknięcia okienka), 10 (kod klawisza kasowania "AC"), ani 15 (kod klawisza "="), przechodzimy do oczekiwania na drugą liczbę, czyli po raz kolejny wywołujemy procedurę "PobierzLiczbe". Gdy już mamy obydwie wartości, możemy wykonać na nich odpowiednie działanie z wykorzystaniem opisanych już wcześniej procedurek. Po tym pozostaje nam już tylko wyświetlenie wyniku na wyświetlaczu (bsr WynikNaWyswietlacz). I to wszystko. W przypadku, gdyby ktoś z Was miał jednak nadal wątpliwości, co do zasady działania programu proponuję uważnie przeczytać komentarze w listingu.

Czekam także na listy z pytaniami i uwagami dotyczącymi tego kursu. W tym miejscu chciałbym uspokoić jednego z czytelników - posiadacza Amigi 1200, który domaga się przedstawienia rozkazów dostępnych na procesorach wyższych od 68000. Nie ma obawy, wszystkie nowe instrukcje i tryby adresowania dostępne dopiero od procesorów 68020 zostaną opisane pod koniec kursu. Dlaczego dopiero pod koniec? Dlatego, że wszystko czego dowiecie się na temat procesora 68000 będzie poprawne dla lepszych procesorów (oczywiście tych z serii 680xx), a to czego nauczylibyście się na temat 68020 i wyżej, byłoby bezużyteczne dla użytkowników A500. ❑

```
;    ## Listing kalkulatora (całość) ##

; ## procedury exec.library ##
_LVOOpenLibrary equ     -552
_LVOCloseLibrary equ    -414

_LVOWaitPort        equ     -384
_LVOGetMsg          equ     -372
_LVOReplyMsg        equ     -378

; ## procedury intuition.library ##
_LVOOpenWindow          equ     -204
_LVOCloseWindow         equ     -72
_LVOWbenchToFront   equ     -342
_LVOWbenchToBack    equ     -336
_LVODrawBorder      equ     -108
_LVOPrintIText      equ     -216

wd_RPort            equ     50
wd_UserPort         equ     86

WINDOWDRAG          equ     2    ; okno można przesuwać po ekranie
WINDOWCLOSE         equ     8    ; okienko będzie miało gadżet
                                 ; służący do jego zamykania
GIMMEZEROZERO       equ     1024 ; współrzędna (0,0) nie będzie w lewym
                                 ; górnym rogu okna, ale w lewym górnym
                                 ; rogu jego części roboczej (wewnętrznej)
WBENCHSCREEN        equ     1    ; okno na ekranie workbencha
GADGHCOMP           equ     0    ; gadżet zmienia kolor po wciśnięciu
GADGIMMEDIATE       equ     2    ; gadżet wysyła komunikat do
                                 ; zaraz po wciśnięciu IDCMP
RELVERIFY           equ     1    ; po puszczeniu przycisku myszy
                                 ; gadżet przejdzie w stan "wyciśnięty"
BOOLGADGET          equ     1    ; gadżet typu wciśnięty/wyciśnięty

CLOSEWINDOW         equ     512  ; system IDCMP będzie reagował na
                                 ; wciśnięcie gadżetu zamykającego okienko
GADGETDOWN          equ     32   ; system IDCMP będzie reagował
                                 ; na wciśnięcie gadżetów

gg_GadgetID         equ     38

im_Class            equ     20
im_IAddress         equ     28

; ## procedury graphics.library ##
_LVOMove        equ     -240
_LVORectFill    equ     -306
_LVOSetAPen     equ     -342

                section Program,code ; poinformowanie assemblera,
                                     ; że teraz będzie program, a nie
                                     ; dane (bez tego też będzie dobrze)

Start:          jmp     Main         ; skok do procedury głównej


; ## Procedury otwarcia i zamknięcia bibliotek oraz okna ##

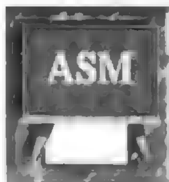
OpenGraphics:
        move.l  4.W,a6                  ; baza biblioteki exec
        lea     GraphicsName(pc),a1     ; nazwa biblioteki graphics
        moveq   #0,d0                   ; nr wersji biblioteki
        jsr     _LVOOpenLibrary(a6)     ; otworzenie biblioteki
        move.l  d0,GraphBase            ; zapamiętanie adresu bazowego
                                        ; biblioteki graphics
        rts

OpenIntuition:
        move.l  4.W,a6                  ; baza biblioteki exec
        lea     IntuitionName(pc),a1    ; nazwa biblioteki intuition
        moveq   #0,d0                   ; nr wersji biblioteki
        jsr     _LVOOpenLibrary(a6)     ; otworzenie biblioteki
        move.l  d0,IntuiBase            ; zapamiętanie adresu bazowego
                                        ; biblioteki intuition
        rts

CloseGraphics:
        move.l  4.W,a6                  ; baza biblioteki exec
        move.l  GraphBase(pc),a1        ; baza biblioteki graphics
        jsr     _LVOCloseLibrary(a6)    ; zamknięcie biblioteki
        rts

CloseIntuition:
        move.l  4.W,a6                  ; baza biblioteki exec
        move.l  IntuiBase(pc),a1        ; baza biblioteki intuition
        jsr     _LVOCloseLibrary(a6)    ; zamknięcie biblioteki
        rts

OpenWindow:
        move.l  IntuiBase(pc),a6        ; baza biblioteki intuition
        lea     WindowDef(pc),a0        ; adres definicji okna
        jsr     _LVOOpenWindow(a6)      ; otworzenie okna
        move.l  d0,WindowBase           ; zapamiętanie adresu
```




```
                                        ; struktury Window
        rts

CloseWdw:
        move.l  IntuiBase(pc),a6        ; baza biblioteki intuition
        move.l  WindowBase(pc),a0       ; adres struktury Window
        jsr     _LVOCloseWindow(a6) ; zamknięcie okna
        rts

WindowDef:  dc.w  200       ; współrzędna x okna
        dc.w    20          ; współrzędna y okna
        dc.w    150         ; szerokość okna
        dc.w    100         ; wysokość okna
        dc.b    0           ; kolor detali w oknie
        dc.b    1           ; kolor wypełnień w oknie
        dc.l    CLOSEWINDOW+GADGETDOWN
                            ; znaczniki IDCMP
        dc.l    WINDOWCLOSE+WINDOWDRAG
                            +GIMMEZEROZERO
                            ; znaczniki określające wygląd i typ okna
        dc.l    Klawisz0GAD             ; adres pierwszej struktury
                                        ; gadżet dla tego okna
        dc.l    0
        dc.l    WindowName; adres testu będącego nazwą okna
        dc.l    0           ; wskaźnik struktury screen dla ekranu na
                            ; którym ma się otworzyć okno (w tym
                            ; przypadku równy zeru, bo używamy
                            ; ekranu Workbencha)
        dc.l    0           ; adres struktury BitMap, jeśli chcemy
                            ; używać własnych bitmap
                            ; (my nie chcemy)
        dc.w    0           ; minimalna szerokość okna
        dc.w    0           ; minimalna wysokość okna
        dc.w    150         ; maksymalna szerokość okna
        dc.w    100         ; maksymalna wysokość okna
        dc.w    WBENCHSCREEN  ; typ okna (w naszym przypadku
                            ; okno na ekranie Workbencha)

WindowName:     dc.b  'Kalkulatorek',0      ; nazwa okna

GraphicsName:   dc.b  'graphics.library',0  ; nazwa biblioteki graphics
IntuitionName:  dc.b  'intuition.library',0 ; nazwa biblioteki intuition

GraphBase:      dc.l  0   ; miejsce przechowywania bazy graphics.library
IntuiBase:      dc.l  0   ; miejsce przechowywania bazy intuition.library
WindowBase:     dc.l  0   ; miejsce przechowywania adresu
                          ; struktury Window dla okienka

WBtoFront:
        move.l  IntuiBase(pc),a6
        jsr     _LVOWbenchToFront(a6)
        rts

WBtoBack:
        move.l  IntuiBase(pc),a6
        jsr     _LVOWbenchToBack(a6)
        rts

;   ## Program główny ##
Main:
        bsr     OpenGraphics    ; wywołanie procedury otwierającej
                                ; bibliotekę graphics
        beq     NoGraphics      ; skok jeśli błąd otwarcia graphics
        bsr     OpenIntuition   ; wywołanie procedury otwierającej
                                ; bibliotekę intuition
        beq     NoIntuition     ; skok jeśli błąd otwarcia intuition
        bsr     WBtoFront       ; ustawienie ekranu Workbencha
                                ; przed innymi ekranami
        bsr     OpenWindow      ; wywołanie procedury
                                ; otwierającej okno
        beq     NoWindow        ; skok gdy błąd otwarcia okna

        bsr     Kalkulator      ; wywołanie głównego podprogramu

        bsr     CloseWdw        ; wywołanie procedury
NoWindow:                       ; zamykającej okno
        bsr     WBtoBack        ; ustawienie ekranu Workbencha
                                ; za innymi ekranami
        bsr     CloseIntuition  ; wywołanie procedury zamykającej
NoIntuition:                    ; bibliotekę intuition
        bsr     CloseGraphics   ; wywołanie procedury zamykającej
NoGraphics:                     ; bibliotekę graphics
        rts                     ; powrót z programu

;  ## Definicje gadżetów klawiszy kalkulatora ##

Klawisz0GAD: dc.l Klawisz1GAD   ; adres następnego gadżetu
        dc.w    5           ; x górnego, lewego rogu
        dc.w    70          ; y górnego, lewego rogu
        dc.w    25          ; szerokość
        dc.w    13          ; wysokość
        dc.w    GADGHCOMP   ; po wciśnięciu gadżetu będzie
                            ; on zmieniał kolor
        dc.w    GADGIMMEDIATE+RELVERIFY ; gadżet będzie
                            ; reagował i na wciśnięcie i na zwolnienie
        dc.w    BOOLGADGET  ; gadżet typu wciśnięty/wyciśnięty
        dc.l    Ramka1,0    ; adres definicji ramki gadżetu
        dc.l    Tekst0      ; adres definicji tekstu w polu gadżetu
        dc.l    0,0         ; MutualExclude i SpecialInfo
                            ; (nie będziemy tego używać)
        dc.w    0           ; numer gadżetu
        dc.l    0           ; nie będziemy tego wykorzystywać

Klawisz1GAD: dc.l   Klawisz2GAD
        dc.w    5,55
        dc.w    25,13
        dc.w    GADGHCOMP,GADGIMMEDIATE
                            +RELVERIFY,BOOLGADGET
        dc.l    Ramka1,0
        dc.l    Tekst1
        dc.l    0,0
        dc.w    1
        dc.l    0
Klawisz2GAD: dc.l   Klawisz3GAD
        dc.w    35,55
        dc.w    25,13
        dc.w    GADGHCOMP,GADGIMMEDIATE
                            +RELVERIFY,BOOLGADGET
        dc.l    Ramka1,0
        dc.l    Tekst2
        dc.l    0,0
        dc.w    2
        dc.l    0
Klawisz3GAD: dc.l   Klawisz4GAD
        dc.w    65,55
        dc.w    25,13
        dc.w    GADGHCOMP,GADGIMMEDIATE
                            +RELVERIFY,BOOLGADGET
        dc.l    Ramka1,0
        dc.l    Tekst3
        dc.l    0,0
        dc.w    3
        dc.l    0
Klawisz4GAD: dc.l   Klawisz5GAD
        dc.w    5,40
        dc.w    25,13
        dc.w    GADGHCOMP,GADGIMMEDIATE
                            +RELVERIFY,BOOLGADGET
        dc.l    Ramka1,0
        dc.l    Tekst4
        dc.l    0,0
        dc.w    4
        dc.l    0
Klawisz5GAD: dc.l   Klawisz6GAD
        dc.w    35,40
        dc.w    25,13
        dc.w    GADGHCOMP,GADGIMMEDIATE
                            +RELVERIFY,BOOLGADGET
        dc.l    Ramka1,0
        dc.l    Tekst5
        dc.l    0,0
        dc.w    5
        dc.l    0
Klawisz6GAD: dc.l   Klawisz7GAD
        dc.w    65,40
        dc.w    25,13
        dc.w    GADGHCOMP,GADGIMMEDIATE
                            +RELVERIFY,BOOLGADGET
        dc.l    Ramka1,0
        dc.l    Tekst6
```

```
                dc.l    0,0
                dc.w    6
                dc.l    0
Klawisz7GAD:    dc.l    Klawisz8GAD
                dc.w    5,25
                dc.w    25,13
                dc.w    GADGHCOMP,GADGIMMEDIATE
                                +RELVERIFY,BOOLGADGET
                dc.l    Ramka1,0
                dc.l    Tekst7
                dc.l    0,0
                dc.w    7
                dc.l    0
Klawisz8GAD:    dc.l    Klawisz9GAD
                dc.w    35,25
                dc.w    25,13
                dc.w    GADGHCOMP,GADGIMMEDIATE
                                +RELVERIFY,BOOLGADGET
                dc.l    Ramka1,0
                dc.l    Tekst8
                dc.l    0,0
                dc.w    8
                dc.l    0

Klawisz9GAD:    dc.l    KasujGAD    ; Tu jest mała, lecz bardzo ważna
        ; zmiana w porównaniu z listingiem z poprzedniego odcinka.
                dc.w    65,25
                dc.w    25,13
                dc.w    GADGHCOMP
                dc.w    GADGIMMEDIATE+RELVERIFY
                dc.w    BOOLGADGET
                dc.l    Ramka1,0
                dc.l    Tekst9
                dc.l    0,0
                dc.w    9
                dc.l    0

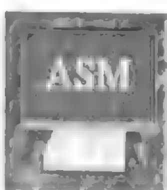
KasujGAD:       dc.l    DzielenieGAD
                dc.w    110,25
                dc.w    25,13
                dc.w    GADGHCOMP
                dc.w    GADGIMMEDIATE+RELVERIFY
                dc.w    BOOLGADGET
                dc.l    Ramka1,0
                dc.l    TekstKasuj
                dc.l    0,0
                dc.w    10
                dc.l    0

DzielenieGAD:   dc.l    MnozenieGAD
                dc.w    110,40
                dc.w    25,13
                dc.w    GADGHCOMP
                dc.w    GADGIMMEDIATE+RELVERIFY
                dc.w    BOOLGADGET
                dc.l    Ramka1,0
                dc.l    TekstDzielenie
                dc.l    0,0
                dc.w    11
                dc.l    0
MnozenieGAD:    dc.l    OdejmowanieGAD
                dc.w    110,55
                dc.w    25,13
                dc.w    GADGHCOMP
                dc.w    GADGIMMEDIATE+RELVERIFY
                dc.w    BOOLGADGET
                dc.l    Ramka1,0
                dc.l    TekstMnozenie
                dc.l    0,0
                dc.w    12
                dc.l    0
OdejmowanieGAD: dc.l    DodawanieGAD
                dc.w    110,70
                dc.w    25,13
                dc.w    GADGHCOMP
                dc.w    GADGIMMEDIATE+RELVERIFY
                dc.w    BOOLGADGET
                dc.l    Ramka1,0
                dc.l    TekstOdejmowanie
                dc.l    0,0
                dc.w    13
                dc.l    0
DodawanieGAD:   dc.l    WynikGAD
                dc.w    80,70
                dc.w    25,13
                dc.w    GADGHCOMP
                dc.w    GADGIMMEDIATE+RELVERIFY
                dc.w    BOOLGADGET
                dc.l    Ramka1,0
                dc.l    TekstDodawanie
                dc.l    0,0
                dc.w    14
                dc.l    0
WynikGAD:       dc.l    0
                dc.w    35,70
                dc.w    41,13
                dc.w    GADGHCOMP
                dc.w    GADGIMMEDIATE+RELVERIFY
                dc.w    BOOLGADGET
                dc.l    Ramka2,0
                dc.l    TekstWynik
                dc.l    0,0
                dc.w    15
                dc.l    0

Ramka1:         dc.w    0,0     ; x,y górnego, lewego rogu
                dc.b    1,0     ; kolor linii
                dc.b    1       ; tryb rysowania (1 to zwykłe
                                ; rysowanie RP_JAM2)
                dc.b    5       ; ile lini
                dc.l    LineTab1 ; adres tabeli współrzędnych punktów
                dc.l    0       ; koniec definicji ramki
LineTab1:       dc.w    0,0     ; x,y pierwszego punktu
                dc.w    24,0    ; x,y drugiego punktu
                dc.w    24,12   ; x,y trzeciego punktu
                dc.w    0,12    ; x,y czwartego punktu
                dc.w    0,0     ; x,y piątego punktu


Ramka2:         dc.w    0,0     ; ramka dla klawisza '='
                dc.b    1,0
                dc.b    1
                dc.b    5
                dc.l    LineTab2
                dc.l    0
LineTab2:       dc.w    0,0,40,0,40,12,0,12,0,0  ; linie dla ramki klawisza '='

Tekst0:         dc.b    1       ; kolor tekstu
                dc.b    0       ; kolor tla
                dc.b    1,0     ; tryb rysowania
                dc.w    8,3     ; x,y górnego, lewego rogu
                dc.l    0       ; adres struktury definiującej rodzaj czcionki
                                ; (gdy 0, to używany jest standardowy font)
                dc.l    Txt0    ; adres właściwego tekstu
                dc.l    0       ; koniec definicji tekstu gadżetu
Tekst1:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt1,0
Tekst2:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt2,0
Tekst3:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt3,0
Tekst4:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt4,0
Tekst5:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt5,0
Tekst6:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt6,0
Tekst7:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt7,0
Tekst8:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt8,0
Tekst9:         dc.b    1,0
                dc.w    0,8,3
                dc.l    0,Txt9,0
```




```
TekstKasuj:     dc.b    3,0
                dc.w    0,5,3
                dc.l    0,TxtKasuj,0
TekstDzielenie: dc.b    1,0
                dc.w    0,8,3
                dc.l    0,TxtDzielenie,0
TekstMnozenie:  dc.b    1,0
                dc.w    0,8,3
                dc.l    0,TxtMnozenie,0
TekstOdejmowanie:
                dc.b    1,0
                dc.w    0,8,3
                dc.l    0,TxtOdejmowanie,0
TekstDodawanie: dc.b    1,0
                dc.w    0,8,3
                dc.l    0,TxtDodawanie,0
TekstWynik:     dc.b    1,0
                dc.w    0,16,3
                dc.l    0,TxtWynik,0

Txt0:           dc.b    '0',0     ; teksty muszą być zakończone zerem
Txt1:           dc.b    '1',0
Txt2:           dc.b    '2',0
Txt3:           dc.b    '3',0
Txt4:           dc.b    '4',0
Txt5:           dc.b    '5',0
Txt6:           dc.b    '6',0
Txt7:           dc.b    '7',0
Txt8:           dc.b    '8',0
Txt9:           dc.b    '9',0

TxtKasuj:       dc.b    'AC',0,0
TxtDzielenie:   dc.b    '/',0
TxtMnozenie:    dc.b    '*',0
TxtOdejmowanie: dc.b    '-',0
TxtDodawanie:   dc.b    '+',0
TxtWynik:       dc.b    '=',0

; ## Procedury kalkulatora ##

Kalkulator:     bsr     RysujWyswietlacz  ; narysowanie ramki wyświetlacza

Czysc1:         bsr     WyczyscWyswietlacz  ; wyczyszczenie wyświetlacza
Czysc2:         move.w  #0,Dzialanie        ; wyzerowanie komórek
                move.l  #0,Liczba1       ; przechowujących rodzaj działania,
                move.l  #0,Liczba2       ; obydwie liczby
                move.l  #0,Wynik         ; i wynik

Dana1:          bsr     PobierzLiczbe    ; pobranie pierwszej liczby
                cmpi.b  #255,d0          ; czy zamknięcie okienka?
                beq     KalkulatorEND    ; jeśli tak, to skok do
                                         ; "KalkulatorEND"
                cmpi.b  #10,d0       ; czy kod klawisza kasowania?
                beq     Czysc1       ; jeśli tak to skok do "Czysc1"
                move.w  d0,Dzialanie   ; zapamiętanie kodu działania (będzie
                                 ; ono wykonane dopiero po pobraniu drugiej liczby)
                move.l  d1,Liczba1     ; zapamiętanie pierwszej liczby

                cmpi.b  #15,d0       ; sprawdzenie, czy aktualnym
                                     ; działaniem jest '='
                bne     Dana2        ; jeśli nie, to skok do pobierania
                                     ; drugiej liczby
                bra     Czysc2       ; jeśli tak, to skok do "Czysc2" jest to
                                     ; zerowanie wszystkich zmiennych
                                     ; bez kasowania wyświetlacza

Dana2:          bsr     PobierzLiczbe    ; pobranie drugiej liczby
                cmpi.b  #255,d0          ; czy zamknięcie okienka?
                beq     KalkulatorEND    ; jeśli tak, to skok
                                         ; do "KalkulatorEND"
                cmpi.b  #10,d0       ; czy kod klawisza kasowania?
                beq     Czysc1       ; jeśli tak to skok do "Czysc1"
                move.l  d1,Liczba2   ; zapamiętanie drugiej liczby
                move.w  Dzialanie(pc),d2   ; zapamiętane wcześniej działanie
                                    ; do wykonania na liczbie1 i liczbie2
                move.w  d0,Dzialanie   ; zapamiętanie nowego
                                       ; działania na przyszłość
                move.l  Liczba1(pc),d0    ; pierwsza liczba do D0
                move.l  Liczba2(pc),d1    ; druga liczba do D1
                cmpi.b  #11,d2            ; sprawdzenie typu
                beq     DzielenieProc     ; działania
                cmpi.b  #12,d2            ; i wykonanie odpowiedniego
                beq     MnozenieProc      ; skoku
                cmpi.b  #13,d2            ;
                beq     OdejmowanieProc;
                cmpi.b  #14,d2            ;
                beq     DodawanieProc     ;
                bra     KalkulatorEND     ; to program nie powinien nigdy
                                          ; dojść, ale nigdy nic nie wiadomo

DzielenieProc:  tst.l   d1
                beq     Blad         ; dzielenie przez zero
                divu    d1,d0
                bvs     Blad         ; błąd występujący przy dzieleniu bardzo
                                     ; dużej liczby przez bardzo małą
                andi.l  #$0000FFFF,d0   ; odcięcie reszty z dzielenia
                                     ; (po DIVU znajduje się ona
                                     ; w starszym słowie wyniku)
                bra     Zrobione1
MnozenieProc:   mulu    d1,d0
                bra     Zrobione1
OdejmowanieProc:sub.l   d1,d0
                bmi     Blad         ; liczba ujemna
                bra     Zrobione1
DodawanieProc:  add.l   d1,d0
                cmpi.l  #99999999,d0  ; większe liczby nie mieszczą
                                      ; się na wyswietlaczu
                bcc     Blad          ; liczba zbyt duża
                bra     Zrobione1

Zrobione1:      move.l  d0,Wynik
                bsr     WynikNaWyswietlacz
                cmpi.w  #15,Dzialanie       ; jeśli nowym działaniem jest '=',
                beq     Czysc2              ; to skok do "Czysc2"
                move.l  Wynik(pc),Liczba1   ; jeśli nowe działanie jest
                bra     Dana2               ; różne od '=', to wynik, staje się
                                ; pierwszą liczbą, a my musimy pobrać tylko
                                ; drugą liczbę i wykonać to działanie

Blad:           lea     WyswietlaczTXT,a0
                move.l  #' ER',(a0)+     ; wpisanie do bufora wyświetlacza
                move.l  #'RROR',(a0)+    ; tekstu " ERROR"
                bsr     WyswietlaczPisz
                bra     Czysc2

KalkulatorEND:  rts                      ; wyjscie z procedury głównej


RysujWyswietlacz:
                move.l  #WyswietlaczRamka,a1
                moveq   #0,d0
                moveq   #0,d1
                move.l  WindowBase(pc),a0   ; adres struktury Window
                move.l  wd_RPort(a0),a0     ; adres struktury RastPort
                                            ; dołączonej do naszego okna
                move.l  IntuiBase(pc),a6    ; baza biblioteki intuition
                jsr     _LVODrawBorder(a6)  ; narysowanie ramki
                rts

WyswietlaczRamka:
                dc.w    5,5
                dc.b    1,0
                dc.b    0
                dc.b    5
                dc.l    WyswietlaczLinie
                dc.l    0

WyswietlaczLinie:
                dc.w    0,0
                dc.w    130,0
                dc.w    130,15
                dc.w    0,15
                dc.w    0,0

CzekajGadget:   move.l  WindowBase(pc),a0      ; adres struktury Window
                move.l  wd_UserPort(a0),a0     ; adres struktury UserPort
                                            ; dołączonej do naszego okna
                move.l  4.W,a6              ; baza biblioteki exec
                jsr     _LVOWaitPort(a6)    ; czekanie na sygnał od IDCMP

                move.l  WindowBase(pc),a0
```




```
        move.l  wd_UserPort(a0),a0
        move.l  4.W,a6
        jsr     _LVOGetMsg(a6)   ; pobranie komunikatu od IDCMP
        tst.l   d0               ; sprawdzenie rejestru D0
        beq     CzekajGadget     ; jednak nie ma komunikatu

        move.l  d0,a1
        move.l  im_Class(a1),d2      ; pobranie typu komunikatu
        move.l  im_IAddress(a1),a2 ; pobranie adresu gadżetu,
                                   ; który wysłał komunikat
        move.l  4.W,a6               ; baza biblioteki exec
        jsr     _LVOReplyMsg(a6)     ; potwierdzenie odebrania

        cmpi.l  #CLOSEWINDOW,d2
        beq     ZamkniecieOkna
        move.w  gg_GadgetID(a2),d0 ; pobranie numeru gadżetu
        rts
ZamkniecieOkna: move.w #255,d0     ; 255 będzie oznaczało wciśnięcie
                                   ; gadżetu zamknięcia okienka
        rts

WyswietlaczPisz:
        moveq   #50,d0      ; współrzędna X tekstu
        moveq   #9,d1       ; współrzędna Y tekstu
        move.l  WindowBase(pc),a0
        move.l  wd_RPort(a0),a0
        move.l  #WyswietlaczTEKST,a1  ; adres definicji tekstu do
                                      ; wyświetlenia
        move.l  IntuiBase(pc),a6      ; baza biblioteki intuition
        jsr     _LVOPrintIText(a6)    ; wyświetlenie zawartości
        rts                           ; wyświetlacza

WyswietlaczTEKST:
        dc.b    1       ; kolor tekstu
        dc.b    0       ; kolor tła
        dc.b    1,0     ; tryb rysowania
        dc.w    0,0     ; x,y górnego, lewego rogu
        dc.l    0       ; adres struktury definiującej rodzaj czcionki
                        ; (gdy 0, to używany jest standardowy font)
        dc.l    WyswietlaczTXT    ; adres właściwego tekstu
        dc.l    0       ; koniec definicji tekstu
WyswietlaczTXT:
        dc.b'       .',0   ; (między pierwszym apostrofem,
                           ; a kropką jest osiem spacji)

WyczyscWyswietlacz:
        lea     WyswietlaczTXT,a0
        move.l  #'    ',(a0)+      ; między apostrofami są 4 spacje
        move.l  #'   0',(a0)+
        bra     WyswietlaczPisz

Bufor1: dc.b    0,0,0,0,0,0,0,0

PobierzLiczbe: lea  Bufor1,a0
        move.l  #0,(a0)+    ; wyczyszczenie bufora liczby
        move.l  #0,(a0)+    ;
        lea     WyswietlaczTXT,a0
        move.l  #'    ',(a0)+; między apostrofami są 4 spacje
        move.l  #'    ',(a0)+; skasowanie bufora wyświetlacza
        lea     Bufor1,a0
PobierzLiczbe1: move.l  a0,-(sp)
        bsr     CzekajGadget
        move.l  (sp)+,a0
        cmpi.b  #255,d0         ; gdy wciśnięty gadżet
        beq     WcisnietyEND1   ; zamknięcia okienka
        cmpi.b  #10,d0          ; gdy wciśnięty gadżet
        beq     WcisnietyKasuj  ; kasowania "AC"
        cmpi.b  #10,d0          ; czy cyfry, czy gadżety
        bcc     KoniecLiczby    ; działań matematycznych
        cmpi.l  #Bufor1+8,a0    ; czy osiągnęliśmy koniec bufora
        beq     PobierzLiczbe1  ; jeśli tak, to nie możemy
                                ; wpisać kolejnej cyfry do bufora
        addi.b  #'0',d0
        move.b  d0,(a0)+        ; wpisanie jednej cyfry do bufora
        move.l  a0,-(sp)
        bsr     BuforNaWyswietlacz  ; wpisanie bufora
                                    ; na wyświetlaczu
        move.l  (sp)+,a0
        bra     PobierzLiczbe1  ; skok do pobierania następnej
                                ; cyfry
KoniecLiczby: cmpa.l #Bufor1,a0 ; czy jest jakakolwiek cyfra ?
        beq     PobierzLiczbe   ; jeśli nie, to nie możemy skończyć
        move.w  d0,-(sp)
        bsr     ASCII_do_liczby ; wywołanie procedury
                                ; przeliczającej kody ASCII na liczbę
        move.w  (sp)+,d0
        rts
WcisnietyKasuj: rts
WcisnietyEND1: rts

BuforNaWyswietlacz:                  ; skopiowanie zawartości bufora1
        suba.l  #Bufor1,a0           ; do bufora wyświetlacza
        move.l  #WyswietlaczTXT+8,a1           ; i wypisanie stanu
                                               ; wyświetlacza
        suba.l  a0,a1
        move.l  #Bufor1,a0
KopiujNaWyswietlacz:
        move.b  (a0)+,(a1)+
        cmpa.l  #WyswietlaczTXT+8,a1
        bne     KopiujNaWyswietlacz
        bsr     WyswietlaczPisz
        rts

ASCII_do_liczby:lea  TabelaMnozen(pc),a1  ; przeliczenie znaków ASCII
        moveq   #0,d1       ; z bufora1 na liczbę
ASCII1: cmpa.l  #Bufor1,a0  ; (w rejestrze D1)
        beq     ASCII4      ; (opis w tekście artykułu)
        moveq   #0,d0
        move.b  -(a0),d0
        subi.b  #'0',d0
        beq     ASCII3
ASCII2: add.l   (a1),d1
        subq.w  #1,d0
        bne     ASCII2
ASCII3: adda.l  #4,a1
        bra     ASCII1
ASCII4: rts                 ; w D1 mamy już właściwą liczbę

TabelaMnozen: dc.l  1,10,100,1000,10000,100000,1000000,10000000

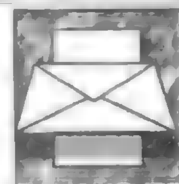
WynikNaWyswietlacz:
        move.l  Wynik(pc),d0        ; wynik do D0
        bsr     Liczba_do_ASCII     ; przeliczenie wyniku na ASCII
        bra     BuforNaWyswietlacz  ; wyświetlenie wyniku

Liczba_do_ASCII:lea  Bufor1,a0          ; przeliczenie liczby zawartej
        lea     TabelaDzielen(pc),a1   ; w rejestrze D0 kody ASCII
        tst.l   d0                     ; (pod adresem "Bufor1")
        beq     ToZERO                 ; (opis w tekście artykułu)
ToASCII1: sub.l (a1),d0
        bcc     ToASCII2
        add.l   (a1)+,d0
        bra     ToASCII1
ToASCII2: moveq #0,d1
ToASCII3: addq.w #1,d1
        sub.l   (a1),d0
        bcc     ToASCII3
        add.l   (a1)+,d0
        addi.b  #'0',d1
        move.b  d1,(a0)+
        cmpa.l  #TabelaDzielenEnd,a1
        beq     ToASCII6
ToASCII4: sub.l (a1),d0
        bcc     ToASCII2
        move.b  #'0',(a0)+
        add.l   (a1)+,d0
        cmpa.l  #TabelaDzielenEnd,a1
        bne     ToASCII4
ToASCII6: rts
ToZERO: move.b  #'0',(a0)+
        rts

TabelaDzielen: dc.l   10000000,1000000,100000,10000,1000,100,10,1
TabelaDzielenEnd:

Dzialanie: dc.w  0   ; tu będzie zapamiętywany kod działania
Liczba1:   dc.l  0   ; tu będzie przechowywana pierwsza liczba
Liczba2    dc.l  0   ; tu będzie przechowywana druga liczba
Wynik:     dc.l  0   ; tu będzie znajdował się wynik operacji
                     ; wykonanej na obydwu liczbach
```

UWAGA! Tekst wytłuszczony należy wpisywać w jednej linii!



Czy istnieją programy [redacted] Amigę, które mogłyby mnie wspomóc przy prowadzeniu małej firmy (rachunki, księgowość, itp.)?

Wbrew pozorom tego typu programy nie są jedynie domeną PC-tów. Należy jednak uczciwie przyznać, że liczba i jakość oprogramowania predysponuje ten komputer do zastosowań biurowych. Jeżeli jesteśmy już posiadaczami Amigi, to nic nie stoi na przeszkodzie, aby zaprząc ją do papierkowej roboty związanej z pracą firmy.

W chwili obecnej najciekawszą i najlepiej opracowaną ofertą legitymuje się krakowska firma "Twin Spark Soft". Proponuje ona następujące tytuły:

- *"Firma"* - program ten jest pomocny przy prowadzeniu księgi podatkowej, rejestrów zakupu i sprzedaży oraz wszelkiego typu ewidencji. Rozliczenia podatkowe uwzględniają podatek VAT.
- *"Kosztorys"* - jak sama nazwa wskazuje, program ten wspomaga obliczenia wszelkiego typu kosztorysów,
- *"Fiskus"* - umożliwia nam prawidłowe obliczanie podatków i wypełnianie związanych z tym deklaracji podatkowych.

Wszystkie wspomniane programy charakteryzują się stosunkowo niską ceną (najbardziej rozbudowany z nich - "Firma", kosztuje ok. 1 mln zł) oraz niewielkimi wymaganiami sprzętowymi (tzn. pracują na wszystkich modelach Amig zaopatrzonych w 1 MB RAM). Zainteresowanym podaję adres

"Twin Spark Soft"
os. Kolorowe 9/16
31-939 Kraków
tel. (012) 44-43-68
fax (012) 47-72-99

Do czego służy "Flicker-fixer" i czy można tego typu urządzenie podłączyć do A500?

Do niedawna "Flicker-fixer" dawał jedyną możliwość dla stosowania monitorów typu multisync i VGA. Umożliwia on bowiem współpracę Amigi z monitorami tych klas, dzięki "podbijaniu" częstotliwości odchylania poziomego do wartości stosowanych w w/w standardach. Pozwala to na korzystanie ze wszystkich trybów graficznych na jednym (znacznie tańszym od multisynców) monitorze. Urządzenie to podłączane może być zewnętrznie do wyjścia RGB komputera (w przypadku A500 - choć bywały również rozwiązania wewnętrzne stosowane w tym komputerze) lub wewnętrznie do odpowiedniego slotu (A2000). Amiga 3000 jest standardowo wyposażona w "Flicker-fixer" (stanowi on integralną część komputera). W przypadku A1200 i A4000, za sprawą nowych kości graficznych, urządzenie tego typu uproszczone zostało do niewielkiej "przejściówki" zmieniającej postać sygnału (między wyjściem RGB, a wejściem VGA). Jeżeli chodzi o drugą część pytania, to istnieje oczywiście wersja tego urządzenia przeznaczona tylko dla A500 (np. "Multi-Vision v.3.0). Cena tej "zabawki" utrzymuje się na poziomie 250-280 DM, co w chwili obecnej przewyższa, praktycznie rzecz biorąc, cenę tego komputera.

Mam komputer podłączony do telewizora "Elemis". Jeżeli przełączę go w "stan oczekiwania" (np. podczas renderingu), to czy komputer może się uszkodzić?

Nawet jeżeli całkowicie odłączymy telewizor od zasilania to nic nie powinno się zdarzyć. Przyczyną uszkodzeń może być jedynie przyłączanie i rozłączanie przewodów połączeniowych, znajdujących się pod napięciem. Zasadniczo dotyczy to monitorów komputerowych, stacji dysków, drukarek i tym podobnych urządzeń. Samo odłączenie zasilania urządzenia peryferyjnego nie wpływa w żaden ujemny sposób na pracę komputera.

Wyłączanie monitora podczas generowania obrazków w programach typu: "Imagine" lub "Real" jest często stosowaną techniką, przedłużającą jego żywotność (w końcu taki "Imagine" potrafi "zamyśleć" się na ładnych parę godzin).

Czy Amigę można zastosować do zadań edukacyjnych (np. nauka matematyki)? Jakie programy najlepiej nadają się do tego celu?

Potężne możliwości graficzne i dźwiękowe Amigi preferują ten komputer do zastosowań edukacyjnych. Do niedawna na naszym rynku oprogramowania dostępne były jedynie obcojęzyczne wersje tego typu programów. W takiej formie nie spełniały one swojej roli. Jednak wraz z upływem czasu i umacnianiem się pozycji Amigi, sytuacja ta uległa zmianie i obecnie istnieje cała ga[redacted] programów edukacyjnych. Można je podzielić na kilka zasadniczych grup:

- edukacja poprzez zabawę - czyli zdobywanie informacji w trakcie gry lub swoistego konkursu - testu. Do tego typu programów należą: "Ortografia" i "Masa" firmy "Twin Spark Soft" oraz "Orto-Test" propagowany przez Fundację Edukacji Technologicznej,
- programy do nauki języka obcego i słowniki. Tutaj najlepszymi przykładami będą: dwuczęściowy "Język angielski" (test + słownik) firmy "TimSoft", "Nauczyciel angielskiego" firmy ASF s.c., "Amisłownik" (polsko-angielski, polsko-niemiecki, angielsko-polski i niemiecko-polski) firmy "Twin Spark Soft", "Tłumacz v2.0" Fundacji Edukacji Technologicznej.
- programy do nauki typowo szkolnych przedmiotów, np.: "Pitagoras" - obejmujący zakres nauki matematyki w szkole podstawowej i średniej, "Europa 2.0" - dotycząca zasad posługiwania się mapą oraz zawierająca podstawowe informacje o poszczególnych państwach Europy,
- oraz jedyny w swoim rodzaju "Supermemo", program pozwalający "nauczyć się uczyć". Jego głównym zadaniem jest bowiem usprawnienie procesu zapamiętywania.

Poza wspomnianymi zastosowaniami można wykorzystać Amigę do wszelkiego typu prezentacji: zdjęć, wykresów funkcji, zasad perspektywy i trójwymiarowego rzutowania, itp.

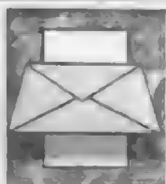
Noszę się z zamiarem kupna "Video Backupu", [redacted] związku z tym mam pytanie: [redacted] ile bezpieczne jest przechowywanie danych za pomoca VBS i od czego zależy jakość takiego zapisu?

Osobiście podchodziłem z dużym sceptycyzmem do VBS-a. Częściowo przekonały mnie wyrażane powszechnie pozytywne opinie o tego typu archiwizacji danych. Daleki jednak jestem od powierzania kasecie video swoich najcenniejszych zbiorów. Zupełnie inaczej ma się sprawa z wszelkiego typu oprogramowaniem PD, grafiką, muzyką itp.

Jakość dokonanego zapisu, a więc i stopień bezpieczeństwa danych zależy przede wszystkim od rodzaju kasety video oraz od stopnia zużycia głowicy nagrywającej w magnetowidzie. Optymalnym rozwiązaniem byłoby zastosowanie firmowej kasety (BASF, TDK, FUJI, inne) [redacted] czasie nagrywania nie przekraczającym 180 minut. W praktyce, jeżeli nie zajdą żadne nieprzewidziane okoliczności, tego typu sposób przechowywania danych jest równie bezpieczny jak stos dyskietek (a o wiele tańszy). Jednak z moich informacji wynika, że jest parę osób, które dzięki VBS plują sobie teraz w twarz.

Czy karty VGA i SVGA mają lep[redacted] parametry od kości AGA?

Ach te porównania. Ciągle dążymy do udowodnienia, że nasz sprzęt jest tym najlepszym. Skoro już takie pytanie się pojawiło, to spróbuję na nie obiek



tywnie odpowiedzieć. Otóż zasadniczo amigowskie kości AGA dają zdecydowanie lepsze możliwości graficzne od kart VGA i SVGA.

Dla standardu VGA maksymalnymi możliwościami graficznymi jest pokazanie 16 kolorów [redacted] palety 262 tys. barw w rozdzielczości 640 na 480 punktów. Analogiczne maksymalne parametry dla SVGA wynoszą: 256 kolorów w rozdzielczości 1024 na 768 punktów (są to wartości teoretyczne dla kart z 1 MB pamięci, w rzeczywistości z trudem stosuje się tryb 256-ścio kolorowy przy rozdzielczości 800 na 600 punktów). Na Amidze ilość dostępnych kolorów w poszczególnych rozdzielczościach jest tylko uzależniona od ilości wolnej pamięci (konkretnie rzecz biorąc, od ilości pamięci typu Chip RAM). Na standardowej A1200, bez dodatkowych rozszerzeń, można pracować w "Deluxe Paint 4.5" z wykorzystaniem trybu HAM-8 (262 tys. kolorów [redacted] teoretycznej 24-bitowej palety) przy rozdzielczości 640 na 512 punktów. Posiadaczom odpowiednich rozszerzeń pamięci kości AGA zaoferują dodatkowe tryby graficzne, począwszy od 800 na 600, aż po 1448 na 566 punktów na ekranie w trybie HAM-8.

Jak się przenosi pliki [redacted] twardych dysków (np. wielkości [redacted] MB) do innego twardego?

Teoretycznie można tego dokonać nawet na zwykłych dyskietkach. Oczywiście ze zrozumiałych względów jest to rozwiązanie mało ekonomiczne.

Najprostrzym sposobem jest podłączenie dwóch twardych dysków ze sobą za pomocą kabelka. Później zostaje już tylko przekopiowanie plików [redacted] docelowy twardy.

Do przenoszenia dużych ilości informacji najlepiej nadają się przenośne dyski twarde -"SyQuesty". Są one podłączane do komputerów przez port SCSI i mają następujące pojemności: 44, 88, 105 i 270 MB.

W uprzywilejowanej sytuacji są również posiadacze stacji CD z możliwością zapisu danych. Wówczas cały problem z przeniesieniem plików sprowadza się do wymiany kompaktu.

Dla mniej zamożnych posiadaczy komputerów ciekawą alternatywą, wobec wspomnianych wcześniej rozwiązań, może być zastosowanie "Video Backupu". Jest to co prawda znacznie mniej "pewny" sposób na przenoszenie danych, ale na pewno najtańszy.

Interesuje mnie współpraca Amigi 500 z magnetowidem. Chodzi mi o nagrywanie na kasetę VHS czołówek-tekstów, animacji itp. Jak podłączyć Amigę i monitor Commodore 1084S z magnetowidem, który posiada tylko dwa wejścia typu CINCH (nie posiada Euro)?

Aby nagrać najprostszą czołówkę video wystarczy zaopatrzyć się w modulator telewizyjny i odpowiednie oprogramowanie (np. "Delux video", "TV-Text", "TV-show"). W A500 podłączenie modulatora odbywa się przez wyjście RGB monitora. Z tego powodu trzeba zrezygnować z tradycyjnego sposobu przyłączania monitora. Postępujemy wówczas następująco: łączymy wejście "Video" magnetowidu [redacted] wyjściem "Video" modulatora, wyjście "Audio" komputera z odpowiednim wejściem magnetowidu, natomiast monitor podłączamy do komputera korzystając z wyjścia "Composite video" i wejścia CVBS (oba typu CINCH).

Programy umożliwiające nagrywanie czołówek na kasetę video są niczym innym, jak odpowiednio przystosowanym "playerem" odtwarzającym stworzoną wcześniej grafikę lub animację pochodzącą z dowolnego programu graficznego (np. "Deluxe Painta"). Istnieje również cały zespół specjalistycznego oprogramowania, jak np. "Scala", "Media Point" czy "Adorage" zajmującego się jedynie tego typu dziedziną tworzenia grafiki.

W najbliższym czasie tematyka poruszona w tym pytaniu zostanie szerzej przedstawiona na łamach "Amigowca".

Czy można wstawić do A600 dysk twardy 3.5"?

Niestety, nie spotkałem się z tym, aby udało się komuś "wcisnąć" dysk twardy 3.5" pod obudowę A600. Podob[redacted] operacja w przypadku znacznie większej A1200 nastręcza już poważne kłopoty, [redacted] co dopiero w przypadku takiego maleństwa jak "sześćsetka". A szkoda. Gdyby się komuś to udało, wówczas A600HD, ze względu na niezwykle niską cenę, mogłaby z powodzeniem powrócić na pierwsze miejsce wśród komputerków domowych (w miejsce 286 i 386SX).

Czy istnieje jakiś dystrybutor sprzedający program "Fantavision" [redacted] instrukcją [redacted] języku polskim?

Kłopoty z oryginalnym oprogramowaniem na Amigę nie przestają być główną bolączką polskich użytkowników tego komputera. Do tej pory jedynie firma IPS podjęła próbę rozpowszechniania zagranicznych programów użytkowych, lecz szybko się z tego wycofała (w zasadzie wszystko rozpoczęło się i skończyło na programie "Deluxe Paint"). Tak więc zakupienie "Fantavision" z polską instrukcją pozostaje jedynie w sferze niespełnionych marzeń. Osoby pragnące żyć w zgodzie z obowiązującym prawem, zmuszone są do pozostawania w kręgu "rodzimej produkcji" lub zakupu oprogramowania za granicą.

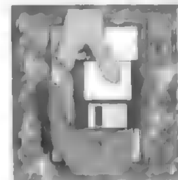
Jaka funkcja ładuje plik danych binarny ?

Do wczytywania danych z dyskietki (gdyż tego chyba dotyczy pytanie) służy funkcja Read z biblioteki dos.library. Jednak, aby móc wczytać plik za pomocą Read, należy wybrany plik otworzyć do czytania wywołując procedurę Open, również z biblioteki dos. Natomiast po zakończeniu czytania danych plik należy zamknąć funkcją Close (też z biblioteki dos).

Przed wywołaniem funkcji Open, w rejestrze D1 należy podać adres pamięci, gdzie jest zapisana nazwa interesującego nas pliku. Nazwa ta musi być zakończona bajtem o zawartości zero. Open wymaga jeszcze zdefiniowania - w rejestrze D2 - czy plik ma być otwarty do pisania, czy do czytania. Jeśli chcemy czytać, to powinna to być wartość MODE_OLDFILE (=1005). Po wywołaniu funkcji Open w rejestrze D0 otrzymujemy wartość tak zwanego FileHandle. Wartość ta jest używana do poinformowania systemu operacyjnego, do którego z otwartych plików (może być ich przecież wiele) chcemy uzyskać dostęp. Jeśli z jakichś powodów jakiegoś pliku o podanej nazwie nie ma na dysku, to w rejestrze D0 będzie znajdowało się zero.

Funkcja Read wymaga [redacted] rejestrze D1 wspomnianego wyżej FileHandle, które otrzymaliśmy z funkcji Open, w rejestrze D2 adresu, pod który dane mają zostać wczytane, [redacted] w D3 - ilości bajtów, jaka ma zostać wczytana. Jeśli nie znamy długości pliku, możemy podać dowolną, dużą liczbę, gdyż wczytane zostanie maksymalnie tyle bajtów, jak długi jest plik. Funkcja Read zwraca w rejestrze D0 rzeczywistą liczbę wczytanych bajtów. Jeśli - przykładowo - podaliśmy, że chcemy wczytać 1000 bajtów, a po wywołaniu Read w D0 znajdowało się 900, to znaczy, że plik był krótszy niż 1000 bajtów i miał długość 900 bajtów.

Dla funkcji Close musimy podać - w rejestrze D1 - FileHandle pliku, który chcemy zamknąć. Zamknięcie pliku jest konieczne, gdyż dopóki tego nie uczynimy, inne programy nie będą mogły ani go skasować , ani do niego pisać. ❑



Public Domain 4/94

Mariusz Ławicki

Dziś, jak co miesiąc, nowy zastrzyk programów, po którym od razu poczujecie się lepiej. Tym razem będzie kilka krótkich programików ułatwiających pracę i zmieniających wygląd Workbencha, lekarstwo dla tych, których denerwuje klikanie stacji oraz jak zwykle coś do zabawy i kilka innych rzeczy. Czytajcie, a znajdziecie!

1. Add Tools V1.11

Użytkownicy Workbencha pracujący na Amigach wyposażonych w Kickstart o numerze nie mniejszym niż 2.0 zauważyli zapewne, że oprócz wielu nowych funkcji nieistniejących w wersji 1.3, na górnej belce pojawiło się nowe menu o krótkiej, ale treściwej nazwie Tools. Jakiż jednak pożytek z takiego wynalazku, jeśli jedyną dostępną funkcją menu Tools jest ResetWB. Wyśmienicie byłoby, gdyby można było umieszczać tam na przykład własne wywołania programów. Czy tak można? Oczywiście, że można, a nawet trzeba po t,o aby maksymalnie uprościć współpracę człowieka z maszyną. Do tego właśnie celu służy program Add Tools. Aby opisać sposób postępowania z tym narzędziem posłużę się prostym przykładem. Załóżmy, że mamy na dyskietce często używany program, który niewiadomo dlaczego jest umieszczony gdzieś głęboko w drzewie katalogów. Aby się do niego "dokopać" trzeba przejść przez masę okienek i szufladek, co przy pracy ze stacją dysków bardziej nerwowych użytkowników może doprowadzić do pasji. Istnieją dwa rozwiązania tego problemu. Można przenieść dany program do katalogu głównego, co nie zawsze jest rozsądnym wyjściem lub dopisać wywołanie programu właśnie do menu Tools. A jak to zrobić? Bardzo prosto. Najpierw należy skopiować program AddTools do katalogu WBStartup: lub dopisać go do sekwencji startowej, a następnie założyć (dowolnym edytorem tekstu) w katalogu S: lub tam gdzie znajduje się program główny plik o nazwie AddToolsData. W pliku tym zawarte będą wywołania programów, zapisane w następującym trzyliniowym formacie:

1 linia - zawiera tekst (nie za długi), który pojawi się w oknie menu. Niech będzie to powiedzmy "Wyświetlacz".

2 linia - tutaj należy podać wywołanie programu, czyli w naszym przypadku na przykład SYS:Utilities/Viewery/Wyświetlacz []. Ten pusty nawias kwadratowy oznacza, że parametrem dla programu Wyświetlacz będzie zaznaczona ikona lub kilka ikon razem (obrazki do przeglądnięcia). W innych przypadkach można oczywiście w ogóle nie wpisywać nawiasu lub podać jakiś stały parametr.

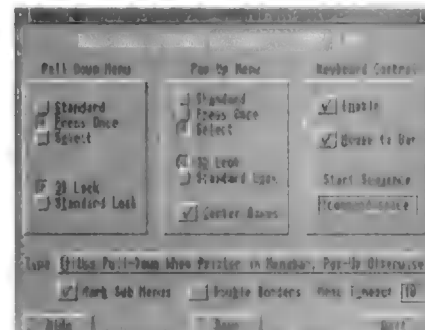
3 linia - w tej ostatniej linii podajemy, w postaci liter S lub A, parametr pracy synchronicznej lub asynchronicznej. O ile wtedy, gdy przy wywołaniu programu zaznaczamy tylko jedną ikonę parametr ten nie ma znaczenia, to w przypadku, gdy zaznaczyliśmy ich więcej, niewłaściwe jego podanie może nam sprawić kłopoty. Praca synchroniczna polega na tym, że podjęcie kolejnego zadania jest możliwe tylko po zakończeniu wykonywania zadania poprzedniego. Przy pracy asynchronicznej wszystkie zadania będą wykonywane równolegle. W naszym przypadku, ponieważ chcemy oglądać obrazki po kolei a nie jednocześnie, powinniśmy wymusić pracę synchroniczną podając S.

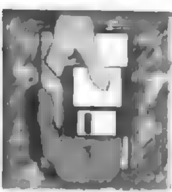
Jeśli wszystkie pliki znajdują się już na właściwym miejscu, to klikamy na ikonę AddTools. Po tym prostym zabiegu ubogie niegdyś menu Tools rozrośnie się o dopisane przez nas opcje, za pomocą których będziemy mogli bezboleśnie uruchamiać nawet najbardziej "zakopane" programy. Gdy zechcemy zrezygnować z usług programu AddTools, wystarczy w menu Tools wybrać opcję AddTools.

Autorem programu jest Alessandro Sala.

2. Magic Menu

Projektanci systemu 2.0 wprowadzili na ekran trochę przestrzenności. Niestety menu pozostało jak za dawnych czasów dwuwymiarowe. Jednak dzięki dużej elastyczności systemu można poradzić sobie z tym problemem i co nieco go uprzestrzennić. Temu właśnie służy ten program. Za jego pomocą można zmienić wygląd i działanie menu. Po uruchomieniu Magic Menu ukazuje się okno z gadżetami służącymi do konfigurowania menu. Oto opis ich znaczenia:





Standard - menu będzie obsługiwane standardowo.

Press Once - menu wywołujemy prawym klawiszem myszki, ale nie trzeba go przytrzymywać podczas wyboru opcji.

Select - tak jak poprzednio, nie trzeba przytrzymywać przycisku myszki w celu dokonania wyboru, ale pod wskaźnikiem myszy nie przesuwa się belka. Przejście do podopcji nie odbywa się automatycznie.

3D Look - uaktywnienie tego gadżetu zmieni wygląd menu na trójwymiarowy.

Standard Look - klasyczne dwuwymiarowe menu

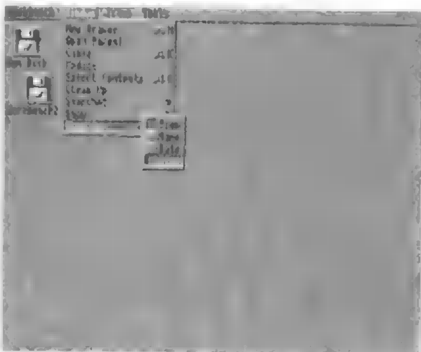
Powyższe gadżety odnoszą się do Pull Down Menu, czyli standardowego górnego menu oraz do alternatywnego Pop Up Menu, które może być wywoływane prawym guzikiem myszy w dowolnym miejscu ekranu. Do tego drugiego typu menu odnosi się jeszcze jeden gadżet, a mianowicie Center Boxes, za pomocą którego można zmienić wzajemne położenie okienek podopcji. Na wygląd menu wpływ mają jeszcze:

Mark Sub Menus - opcje menu posiadające podopcje będą dodatkowo oznaczane.

Double Borders - okna menu będą miały podwójne ramki.

Z menu można korzystać nie tylko przy pomocy zwierzaka, ale także za pomocą klawiatury. Odpowiednich ustawień dokonać można używając gadżetów Keyboard Control:

Enable - włączenie umożliwia obsługę menu z klawiatury.



To jest Pull Down Menu.

Mouse To Bar - przy "klawiszowej" obsłudze menu kursor myszy jest przenoszony do górnej belki. Jest to ważne w niektórych programach (np. DPaint).

Start Sequence - w okienko pod tym napisem wpisujemy kombinację klawiszy, za pomocą której będziemy uaktywniać menu. Oto spis dostępnych możliwości:

lshift, rshift, shift - lewy, prawy lub dowolny klawisz Shift.

lalt, ralt, alt - jak wyżej, tylko w odniesieniu do klawisza Alt.

control - klawisz Ctrl.

lcommand, rcommand - lewa lub prawa Amiga.

up, down, right, left - klawisze kursora.

numericpad - klawiatura numeryczna.

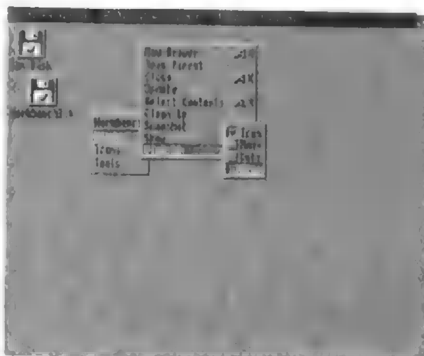
return, enter - to prawie to samo, z tą różnicą, że ten drugi znajduje się w bloku numerycznym.

f1 do f10 - klawisze funkcyjne.

space, backspace, tab, esc, del, help, capslock - bez komentarza.

To tyle co do klawiatury. Warto jeszcze dodać, że za pomocą gadżetu Type możemy ustalić, z jakich rodzajów menu będziemy korzystać. Do wyboru mamy:

Use Pull-Down When Pointer In Menu Bar, Pop-Up Otherwise - kiedy kursor myszy będzie znajdował się na obszarze górnej belki, używane będzie górne menu, w przeciwnym wypadku, po kliknięciu prawym klawiszem myszki menu będzie otwierane w aktualnym położeniu wskaźnika.



A tak wygląda Pop Up Menu.

Use Pull-Down Menu Only - będzie używane tylko górne menu.

Use Pop-Up Menu Only - do dyspozycji będziemy mieli tylko alternatywne menu dostępne w każdym miejscu ekranu.

Nieco więcej możliwości, jak na przykład ustalenie priorytetu, można podać bezpośrednio do Tooltypes ikony programu, ale ponieważ opis dodatkowych funkcji zająłby zbyt wiele miejsca, zainteresowanych zachęcam do przeczytania "doca".

Kiedy już ustawiliśmy wszystkie parametry, możemy je zapisać klikając na gadżet Save. Konfiguracja jest zapisywana do Tooltypes ikony programu, a jeżeli takowej nie ma, to zostanie automatycznie założona. W przypadku kiedy nie chcemy nagrywać ustawień, wystarczy kliknąć na gadżet Hide. Program jest rezydentny i po aktywacji w każdej chwili możemy na nowo wywołać okno konfiguracyjne, za pomocą kombinacji klawiszy Ctrl+Alt+Space. Jeśli chcemy zrezygnować z nowego menu, wywołujemy okno z gadżetami i klikamy na Quit. Po tej operacji zostanie otworzone nowe okno, a do wyboru będziemy mieli:

Deactivate - deaktywacja nowego menu, ale bez wyjścia z programu.

Force Uninstall - wyłączenie menu, połączone z wyjściem z programu.

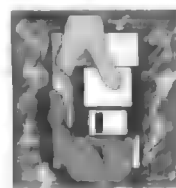
Cancel - rezygnacja.

Jeżeli będziecie chcieli zainstalować ten program na innych dyskietkach, wystarczy umieścić go w katalogu WBStartup lub jeśli nie korzystacie z Workbencha, skopiować MagicMenu na dyskietkę i dopisać go do sekwencji startowej. Myślę, że nowe przyjemnie przestrzenne i wygodne w pracy menu bardzo Wam się spodoba. I jeszcze drobna uwaga program wymaga systemu 2.0 lub nowszego.

Autorem programu jest: Martin Korndorfer.

3. Toggle Click

Tomek był normalnym chłopakiem, jakich miliony na Ziemi. Kiedy rodzice kupili mu Amigę był najszczęśliwszym z ludzi. Wydawać by się mogło, że to sielanka, ale coś dziwnego wprost zżerało Tomka od środka, robił się coraz



chudszy i coraz bardziej nerwowy. Nie pomagały wizyty u lekarzy i cudotwórców. Przyczyna coraz bardziej pogarszającego się stanu Tomka była wciąż nieznana. W końcu zespół wybitnych specjalistów ustalił źródło choroby: wszystkiemu winne było KLIKANIE STACJI. Tomkowi przepisano cudowny lek o nazwie Toggle Click, który należało zażywać po każdym włączeniu komputera. Od tego czasu Tomek jest zdrów jak przysłowiowa rybka.

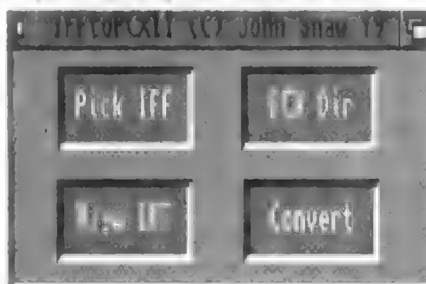
Także Ty, jeśli nie chcesz cierpieć użyj Toogle Click. Wśród innych produktów tego typu program ten wyróżnia się bardzo małą objętością, ma zaledwie 444 bajty! Po jego uruchomieniu uciążliwe klikanie zniknie. Jeśli chcemy powrócić do poprzedniego stanu, musimy po prostu uruchomić Toggle Click ponownie. Program uruchamia się z poziomu CLI z podaniem numeru lub numerów napędów, które nie mają klikać. Jeśli chcemy za jednym zamachem przeprowadzić tę operację dla wszystkich stacji, parametrem musi być KILLALL. Mogę dodać jeszcze, że programik działa na maszynach wyposażonych w Kickstart 2.04. I zapanowała cisza...

Autorem programu jest Sebastiano Vigna.

4. WBSearch

To prosty i bardzo pożyteczny program służący do wyszukiwania plików. Po uruchomieniu z pozoru nic się nie dzieje, jednak po bliższym przeglądzie systemu zauważymy, że w menu Tools Workbencha przybyła opcja Search. Jak tego używać? Bardzo prosto. Wystarczy pojedynczym kliknięciem wybrać napęd lub katalog (można kilka), w którym ma być odszukany dany plik, a następnie skorzystać z opcji Search umieszczonej w menu Workbencha. Po tych operacjach otrzymamy okienko z miejscem na wpisanie nazwy pliku do znalezienia, a następnie otworzy się drugie okno, w którym na bieżąco można będzie kontrolować efekty wyszukiwania oraz przerwać pracę przez kliknięcie na Stop. Podaną do znalezienia nazwę program traktuje jako fragment, to znaczy, że jeżeli zechcemy znaleźć na dysku plik Dir, to oprócz niego zostaną wyszukane wszystkie pliki zawierające ten ciąg znaków (np. MakeDir). Możliwa jest praca z kilkoma jednocześnie wywołanymi poszukiwaczami. Trzeba tylko uważać, aby nie pracowały one na tym samym napędzie, gdyż powoduje to nieprzyjemne efekty, objawiające się głośnym warkotem głowicy skaczącej po sektorach. Wyjścia z programu dokonujemy przez wybranie opcji Search bez zaznaczenia miejsca poszukiwania. Programik ten działa na maszynkach wyposażonych w system 2.0 lub wyższy i może oddać nieocenione usługi, zwłaszcza przy pracy z twardym dyskiem.





Autorem programu jest Mike Austin.

5. Idle Led

To programik, który umożliwia optyczną kontrolę obciążenia procesora. Jego działanie polega na wygaszaniu diody POWER komputera wtedy, gdy CPU jest wolny oraz zaświecaniu jej, kiedy nasza Motorolka ma zajęcie. Objawia się to sporadycznym mruganiem ledki, gdy nie wykonywane są żadne zadania. Im procesor będzie bardziej zajęty, tym z większą częstotliwością dioda będzie się świecić.

Autorem programu jest Lindsay Meek.

6. IFF to PCX

Jeśli ktoś dokonuje konwersji grafiki, to zazwyczaj przerabia ściągnięte od kolegi z pecetem skanowane obrazki jakich pełno u niebieskich. Czasami istnieje potrzeba wykonania rzeczy odwrotnej, czyli przetransformowania obrazków formatu IFF z Amigi, na jakiś bardziej zrozumiały dla peceta. Takie właśnie zadanie spełnić może ten program. Za jego pomocą można dokonać konwersji z IFFa na format PCX dosyć często wykorzystywany przez pecetowe programy do obróbki grafiki.

Obsługa programu jest prosta i sprowadza się do klikania na gadżety. Najpierw za pomocą Pick IFF należy wczytać do pamięci grafikę w formacie IFF. Tutaj pojawia się pewne ograniczenie, a mianowicie liczba kolorów kowertowanych obrazków nie może przekraczać 16. Po załadowaniu obrazka możemy go sobie obejrzeć klikając na View IFF. Kolejną czynnością jest podanie ścieżki dostępu dla pliku PCX, aby to zrobić naciskamy na PCX Dir. Kiedy mamy załadowaną grafikę i podaną ścieżkę dostępu możemy klikając na gadżet Convert przystąpić do punktu kulminacyjnego czyli do konwersji. Po krótkiej chwili w żądanym miejscu powinien znaleźć się plik o takiej samej nazwie jak plik wejściowy, wzbogacony tylko o rozszerzenie PCX. Jeśli długość nazwy pliku IFF przekraczała 8 znaków, to zostanie odpowiednio skrócona. I to byłoby tyle. Przytoczę jeszcze tylko stare ludowe porzekadło: "A ja wolę Motorolę!!!". Tak trzymać.

Autorem programu jest John Shaw.

7. Planetarium V1.0

Niektórzy ludzie wierzą, że układ gwiazd i planet ma olbrzymi wpływ na życie człowieka, inni zaś twierdzą, że to bajeczki dla niegrzecznych dzieci. Niezależnie od tego, która ze stron ma rację czasem może nas ciekawić jak właśnie teraz ustawione są planety. Aby się tego dowiedzieć nie trzeba wertować książek i męczyć kalkulatorów. Wystarczy mieć komputer i program Planetarium. Właśnie za jego pomocą możemy zobaczyć układ planet w wybranym przez nas dniu. Po uruchomieniu programu na ekranie ujrzymy układ słoneczny oraz spis opcji dostępnych z klawiatury. To jest:

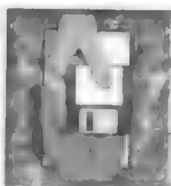
F1 Change Date - tutaj ustawiamy wybraną datę, podajemy ją w formacie: miesiąc/dzień/rok.

F2 Current Date - data będzie ustawiona zgodnie ze stanem zegara systemowego.

F5 Animate On - włączenie opcji animacji ruchu planet.

F6 Animate Off - funkcja odwrotna do powyższej.

F7 Forward - podczas animacji data będzie zwiększana.



F8 Reverse - funkcja odwrotna do powyższej.

F9 Slow - powolna animacja.

F10 Fast - animacja szybka (duże przeskoki czasowe).

Help - informacje o programie.

Esc - wyjście.

Program jest bardzo przyjemny, ale ma jedno poważne ograniczenie, mianowicie podawana data musi się mieścić w zakresie od 1990 do 1995 roku. Autor obiecuje wydanie rozszerzonej wersji bez tego ograniczenia i jeżeli wpadnie mi ona w ręce, nie omieszkam się z Wami podzielić.

Autorem programu jest Jim Schwartz.

8. MiniPac V1.0

Pacman, czyli zjadacz kropek to gra stara jak komputery, i nie ma chyba takiego, choćby najbardziej nietypowego komputera dla którego nie powstałaby implementacja tej zabawy. Na naszą przyjaciółkę również powstało wiele odmian Pacmana, a to właśnie jedna z nich. Nasza wersja to wersja mini, która mieści się w okienku mniejszym niż jedna czwarta ekranu, ale mimo to (pomijając brak dźwięku) zachowuje się jak duży Pacman. Po uruchomieniu ukazuje się okno z takimi oto gadżetami:

Iconify - usypia program i zamienia okienko w krótką belkę. Powrotu dokonujemy przez kliknięcie na belce najpierw lewym, a potem prawym guzikiem myszki.

About - informacje [redacted] programie i jego autorze.

Prefs - ustawianie konfiguracji. Po kliknięciu [redacted] ten gadżet otwiera się nowe okno, w którym możemy ustalić:

Level - poziom trudności (szybkość gry).

Autopause - pauza będzie deaktywowana przez naciśnięcie klawisza lub ruch joysticka.

Priority - priorytet.

Sprite Colors - tu ustawiamy kolory duszków. Możemy wybierać między jasnymi (Bright) i ciemnymi (Dark).

Smooth Moves - włączenie lub wyłączenie płynnego ruchu duszków.

Ignore - do wyboru mamy dwa gadżety oznaczone jako Port 1 oraz Port 2. Od ich ustawienia zależy, [redacted] którego portu joysticka będzie korzystał program. Gadżet ustawiony oznacza pominięcie portu.

Tyle [redacted] ustawieniach. W oknie głównym znajdują się jeszcze gadżety Start (rozpoczęcie gry) i Pause (przerwa w grze). Wyświetlana jest także liczba zdobytych punktów oraz ilość istnień kropkojada. A teraz nie trać już czasu, łap za drąga albo wal w klawisze (byle nie za mocno!) i nakarm wiecznie głodnego stworka.

Autorem programu jest Philippe Banwarth.

9. Minisola V1.0

Tak jak Minipac jest to gra typu mini i jak to w takich grach bywa, wszystko jest w niej mini. Wszystko oprócz myślenia, które powinno być raczej ustawione na maxi, a to z tego względu, że w odróżnieniu od wspomnianego Minipaca, który niewątpliwie należy do gier zręcznościowych, Minisola jest grą logiczną. Zanim opiszę zasady gry, przyjrzymy się dostępnym gadżetom.

Iconify - uśpienie programu.

About - informacje o programie i autorze.

Prefs - konfiguracja. Możemy tutaj ustawić wielkość pola gry, podając dla współrzędnej Lines liczbę z zakresu od 4 do 10 oraz dla współrzędnej Columns liczbę z zakresu od 5 do 10. Oprócz wymiarów planszy można ustalić, który z graczy będzie rozpoczynał grę, ustawiając gadżet First to play na Alternate (pierwszeństwo losowane) lub Player (gracz) albo też Computer.

Start - rozpoczęcie rozgrywki.

Gra toczy się na prostokątnym lub kwadratowym polu, po którym poruszają się gracze. Najpierw wykonujemy ruch na dowolne sąsiednie pole, [redacted] następnie niszczymy jedno wybrane pole na planszy. [redacted] to pole nie może wejść już żaden gracz. Postępując w ten sposób dążymy do takiej sytuacji, w której przeciwnik nie może wykonać ruchu. Oczywiście przez cały czas, oprócz atakowania komputera musimy uważać na wykonywane przesunięcia, aby nie dać zapędzić się w miejsce, [redacted] którego nie wyjdziemy. Wygrywa ten, któremu pierwszemu uda się zablokować ruchy przeciwnika. Wszystko jasne? Jeśli tak, to zapraszam do gry. Pomimo swojej prostoty jest naprawdę wciągająca i zmusza do myślenia.

Autorem programu jest Philippe Banwarth.

10. Listingi programów.

W tej szufladzie znajdziecie listingi do wszystkich kursów języków programowania prowadzonych w Amigowcu.

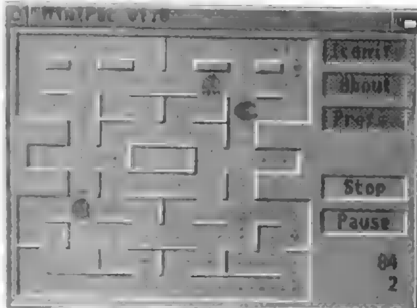
Autorem większości ikonek zamieszczanych [redacted] naszych dyskietkach jest Dariusz Zwierzyński.

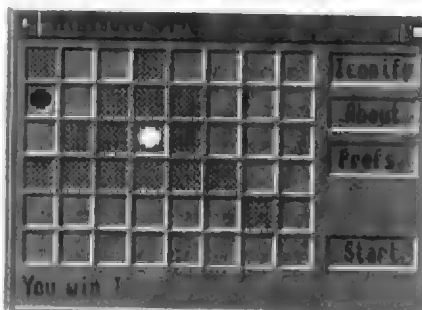
Prosimy o uwagi o naszej bibliotece. Piszcie do nas, jakich programów poszukujecie i jakie powinny się w niej znaleźć. Czekamy też na owoce Waszej pracy.

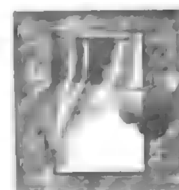
* * *

Jak zamawiać i ile to kosztuje?

Dyskietki PD można zamawiać przesyłając pieniądze na nasze konto, przy pomocy blankietu z Amigowca, podając symbol dysku PD (od #1/92 do 12/92, #1/93, #2/93, #3-4/93, #5-6/93, #7-8/93, #1/94, #2/94, #3/94, 4/94 oraz "A", "B" i "C") i oczywiście swój czytelny adres. Nasza dyskietka kosztuje 48 tysięcy złotych, [redacted] dla prenumeratorów 38 tysięcy, (wliczone są koszty dyskietki, przesyłki, nalepki, opakowania itp.). W przypadku przesyłki za zaliczeniem pocztowym, zamawiający pokrywa koszty zaliczenia. ❑







LISTA DEMONÓW

Artur Łukasik, Sebastian Kłomski

Witamy po miesięcznej nieobecności! Na zachodniej scenie stagnacja trwająca od PARTY III w Herning w Danii (czyli od grudnia), ale za to na polskiej szykują się wielkie zmiany. Spowodowane jest to zapewne wiosennym zlotem polskiej sceny, który odbył się w Starachowicach. Od tego numeru postanowiliśmy wprowadzić kilka technicznych zmian. Główna polega na zwiększeniu liczby miejsc w najważniejszych rubrykach naszej listy.

ZAGRANICA

GRUPY:

1. *Sanity*
2. *Virtual Dreams/Fairlight*
3. *Kefrens*
4. *Razor 1911*
5. *Complex*
6. *Spaceballs*
7. *Melon Dezign*
8. *Movement*
9. *Lemon*
10. *Silents*
11. *Pygmy Projects*
12. *Andromeda*
13. *Digital*
14. *Reflect*
15. *Balance*
16. *Dual Crew & Shining*
17. *Phenomena*
18. *TRSI*
19. *Static Bytes*
20. *Majic 12*

Najlepszy koder na świecie, czyli Chaos z niemieckiej grupy Sanity, od dłuższego czasu zapowiadał wspaniałe demo na nasz ukochany komputer. Dowiedzieliśmy się jednak, że przez kilka najbliższych miesięcy demo to nie ujrzy światła dziennego. Szkoda, ponieważ grupy Virtual Dreams/Fairlight i Razor 1911 pracują nad nowymi szokującymi produkcjami, dzięki którym mogą odebrać grupie Sanity palmę pierwszeństwa. Podejrzewamy, że osamotniony Laxity z grupy Kefrens (szczegóły w poprzednim Amigowcu) również szykuje niespodziankę na najbliższe party w Danii. Jego byli towarzysze z grupy (czyli Vention-koder i R.W.O-grafik) założyli grupę Polka Brothers, lecz na razie nic o nich nie słychać.

DEMA:

1. **Hardwired/***Cronics & The Silents*
2. **Arte/***Sanity*
3. **Desert Dream/***Kefrens*
4. **Origin/***Complex*
5. **Full Moon/***Fairlight*
6. **Extension/***Pygmy Projects*
7. **9 Fingers/***Spaceballs*
8. **World of Commodore/***Sanity*
9. **Groovy/***Lemon*
10. **Dream Tripping/***Digital*
11. **State of The Art/***Spaceballs*
12. **How to skin a cat** */Melon Dezign*
13. **Guardian Dragon II/***Kefrens*
14. **D.A.N.E./***Kefrens*
15. **Sound Vision/***Reflect*
16. **Guardian Dragon I/***Kefrens*
17. **Delirium/***Complex*
18. **Odyssey/***Alcatraz*
19. **Enigma/***Phenomena*
20. **Global Trash/***The Silents*

DEMA PLIKOWE:

1. **Interference/***Sanity*
2. **3D Demo II/***Anarchy*
3. **Mindriot/***Andromeda*
4. **Paradigma/***Complex*
5. **Announce/***Lemon*
6. **Transpiersky/***The Silents*
7. **Clairvoyance/***Absolute*
8. **Terminal Fuckup/***Sanity*
9. **Blue House II/***Rebels*
10. **Joyride/***Phenomena*

INTRA 40 KB:

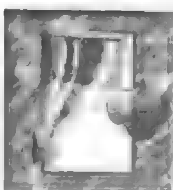
1. **Tetris/***Melon Dezign*
2. **Virtual Dreams**
3. **Arrilfoll/***Dual Crew & Shinig*
4. **Shining**
5. **Fairlight**
6. **Lemon**
7. **Stellar**
8. **The Silents**
9. **Spaceballs**
10. **No Pain No Gain/***TRSI*

MUSIC DISKI:

1. **Memorial Songs/***Alcatraz*
2. **Jesterday/***Sanity*
3. **Chromagic** */Dual Crew & Shining*
4. **Turmoil/***Sanity*
5. **Mirror/***Andromeda*
6. **Crystal Symphonies II** */Phenomena*
7. **Kuglep0len/***Static Bytes*
8. **Music Dream II/***Phenomena*
9. **Book of Song/***Complex*
10. **Crystal Symphonies I** */Phenomena*

KODERZY:

1. **Chaos/***Sanity*
2. **Laxity/***Kefrens*
3. **Mr.Pet/***Sanity*
4. **Tsunami/***Fairlight*
5. **The Spy/***Crionics*
6. **Lone Starr/***Spaceballs*
7. **Dr.Skull/***Fairlight*
8. **Performer/***Melon Dezign*
9. **Microforce/***Sanity*
10. **Gengis/***Complex\f\ff<(CG) HelveticaPL-Regular>*



GRAFICY:

1.**Cougar/*Sanity***
2.**Ra/*Sanity***
3.**Facet/*Lemon***
4.**Mack/*Melon Dezign***
5.**R.W.O./*Polka Brothres***
6.**Peachy/*Masque***
7.**Hof/*Melon Dezign***
8.**Walt/*Melon Dezign***
9.**Rack/*Absolute***
10.Titan/***Complex***

MUZYCY:

1.**Lizard King/*Razor 1911***
2.**Jester/*Sanity***
3.**Moby/*Sanity***
4.**Audiomonster/*Melon Dezign***
5.**Laxity/*Kefrens***
6.**Virgill/*Masque***
7.**Nuke/*ex. Lemon***
8.**Clawz/*Complex***
9.**Jesper Kyd/*The Silents***
10.**Tip [redacted] Mantronix/*Phenomena***

POLSKA

GRUPY:

1.**Old Bulls**
2.**Deform**
3.**W.F.M.H.**
4.**Applause**
5.**Alchemy**
6.**Flying Cows Inc.**
7.**Mystic PL**
8.**Suspect**
9.**Beta Team**
10.**TRSI PL**
11.**Damage**
12.**Illusion**
13.**Union**
14.**Freezers**
15.**Joker**
16.**Blaze**
17.**Lady Bird Design**
18.**Investation**
19.**Funzine**
20.**Obsession**

Dopiero po wprowadzeniu dwudziestu miejsc widać jak na dłoni sytuację [redacted] krajowej scenie. Poniżej dziesiątego miejsca znajduje się wiele młodych grup, które nigdy przedtem nie były w żadnym notowaniu. Poniżej przedstawiamy najświeższy skład wrocławskiej grupy Alchemy:

Axel - kodowanie
Backfire - swapowanie
Bob - kodowanie
Cromax - grafika
Exolon - swapowanie
Johan - grafika
Ozzy - grafika
Saggitarius - grafika
TipTop - kodowanie
Vico - kodowanie

Ponadto grupa ta posiada zagraniczną sekcję w Norwegii:

The Clash - muzyka
Walker - swapowanie

DEMA:

1.**Impulse 2/*TILT/Old Bulls***
2.**Ray World/*Deform***
3.**PRO.s.i.a.k/*FCI***
4.**Hallucinations [redacted] Dreams/*Union***
5.**Deformations/*Deform***
6.**Fugazi/*Old Bulls***
7.**Software/*W.F.M.H.***
8.**Marchewki/*Alchemy***
9.**Technological Death/*Mad Elks***
10.**The Return/*Joker***
11.**Overvision/*Blaze***
12.**Uquala/*Damage***
13.**Faster than Hell/*W.F.M.H.***
14.**Toxic Ziemniak/*Alchemy***
15.**Weeken Dance/*Deform***
16.**Hallucinations [redacted] Dreams preview/*Katharsis***
17.**Hydra/*Investation***
18.**Hexadecimal Perfection /*Action Direct***
19.**Damage/*Old Bulls***
20.**Alternative Reality/*Suspect***

Po dodaniu dziesięciu kolejnych pozycji widać jak uboga w tej dziedzinie jest polska scena. Zauważmy, że prawie jed[redacted] trzecia polskich dobrych dem to produkcje dwóch czołowych grup Old Bulls i Deform.

DEMA PLIKOWE:

1.**Party-zanci/*Joker***
2.**Bara Bara/*Applause***
3.**40-k intro/*Alchemy***
4.**Impulse-preview/*Old Bulls***
5.**Varathron/*Suspect***
6.**Autobiography/*Old Bulls***
7.**X-Mas/*Damage***
8.**Poison/*Ivestation***
9.**No Name/*Suspect***
10.**Xenium/*Old Bulls***

DYSKI MUZYCZNE:

1.**Orgasm/*Old Bulls***
2.**Waiting for Another Beer /*Joker***
3.**Vengeance/*Beta Team***
4.**Hunt for the red kret/*FCI***
5.**Freedom for Lady Birds**
6.**Kadi Sound Disk III/*Joker***
7.**Illusion/*Old Bulls***
8.**Music Tracks/*Suspect***
9.**Lost Tracks/*Illusion***
10.**Techno Fuss/*Investation***

KODERZY:

1.**Musashi/*Union***
2.**Hudi/*TILT/Old Bulls***
3 **Bukka/*TILT/Old Bulls***
4.**Robin/*W.F.M.H.***
5.**Falcon/*Applause***
6.**Vico/*Alchemy***
7.**Tom/*TRSI PL***
8.**Dak/*Mad Elks***
9.**Miklesz/*Damage***
10.**Kane/*Suspect***

MUZYCY:

1.**XTD/*Union***
2.**Peters/*Joker***
3.**Pic Saint Loup**
4.**Accord/Gel Dezign /*Old Bulls***
5.**Dreamer/*FCI***
6.**Mr.Root/*Union***
7.**Snoopy/*Energy***
8.**Phobos/*Beta Team***
9.**Glue/*Damage***
10.**Kadi/*Joker***

GRAFICY:

1.**TPP/*Applause***
2.**Animal/*TRSI PL***
3.**Frodo/*Alchemy***
4.**Zefir/Gel Dezign/*Old Bulls***
5.**Seq/*TRSI PL***
6.**Sabe/*TILT/Old Bulls***
7.**Pluton/*TRSI PL***
8.**Rys/*Joker***
9.**Python/*TRSI PL***
10.**Berserker/*Investation***

MAGAZYNY DYSKOWE:

1.**Thing/*Applause***
2.**Zasmashka/*Alchemy***
3.**X-mag/*Union***
4.**Poczytaj Mi Mamo**
5.**Paper White/*Luzers***
6.**Zig Zag/*Union***
7.**Nie z tej beczki/*Lamers***
8.**Fat Agnus/*Investation***
9.**Krawężnik/*Status OK***
10.**Imazine/*Saint Group***

Wszystkie wiadomości pochodzą z pewnych źródeł i są w 100 procentach sprawdzone.

Przypominamy, że można głosować na dowolną ilość pozycji z każdego działu, przyznając im od 1 do 10 punktów.

Głosy, propozycje, swoje prace, uwagi, [redacted] także najnowsze programy demonstracyjne i magazyny dyskowe prosimy przesyłać na adres redakcji. ❑



Tworzenie własnego MENU

Menu rozwijane z górnej belki można by uznać [redacted] typowy element programów dla komputera Amiga. Rozwiązanie takie umożliwia łatwe i szybkie wybieranie poszczególnych opcji w programie, pozwalając nawet przy znacznej ich ilości na zachowanie porządku poprzez pogrupowanie dostępnych funkcji.

Pisząc w Amosie bardziej rozbudowane programy warto zdecydować się na takie właśnie menu, choćby ze względu na panującą w tym zakresie standardyzację programów i bogaty zestaw komend, przy pomocy których w łatwy sposób można typowe menu zdefiniować.

Amos daje programiście dużą swobodę [redacted] definiowaniu menu. Mogą one składać się [redacted] maksymalnie 8 poziomów i każde menu jednostkowe może być swobodnie przemieszczane po ekranie zarówno przez użytkownika, jak i przez instrukcje sterujące umieszczone [redacted] programie. Wygląd takiego menu możemy dostosować do swoich potrzeb i gustu poprzez zmianę kolorów, zestawu znaków oraz umieszczenie bobów i ikon. Jeśli dodamy do tego możliwość uruchamiania poszczególnych opcji przez wciśnięcie wybranego klawisza (lub ich kombinacji), to [redacted] całą odpowiedzialnością można stwierdzić, że menu stworzone [redacted] Amosie będzie prezentować wysoki poziom i z pewnością nie będzie ustępować tworzonym w innych językach programowania. Czas zatem na konkrety, czyli przedstawienie odpowiednich komend, popartych oczywiście stosownymi przykładami.

Menu$(a,b,c...)=t$

a,b,c - numery pozycji menu

t$ - zmienna tekstowa zawierająca tytuł menu (opcji)

Funkcja ta pozwala na zdefiniowanie podstawowego menu. Zmienna t$ musi zawierać tekst, który ma później ukazywać się jako nazwa menu lub opcji. Odpowiedni dobór wartości zmiennych a,b,c .. pozwala na definiowanie poszczególnych części menu. Jeżeli w nawiasie umieścimy tylko jedną liczbę, to podany tekst będzie stanowił tytuł menu pojawiający się na górnej belce. Tytuły te numerowane są kolejno od lewej do prawej, poczynając od jeden. np.

Menu$(1)=" Amiga "

Instrukcja taka umieści z lewej strony górnej belki napis "Amiga" stanowiący tytuł pierwszego menu (rozwijanego w dół).

Menu$(2)=" Amos "

A w ten sposób z prawej strony poprzedniego tytułu umieścimy tytuł drugiego menu. Wstawiając w nawias liczbę 3 zdefiniujemy tytuł trzeciego menu itd. Trzeba pamiętać [redacted] spacjach przed nazwą menu, aby poszczególne wyrazy nie "zlały się" w jeden przy wyświetlaniu na górnej belce. Jeśli po liczbie określającej nu[redacted] menu podamy drugą liczbę, to będzie [redacted] określać numer opcji w danym menu, a tekst zawarty w zmiennej będzie stanowił jej nazwę. Opcje numerowane są od góry do dołu zaczynając od 1.

Menu$(1,1)=" [redacted] "

Taka instrukcja zdefiniuje pierwszą opcję pierwszego menu. Podając kolejne numery opcji (druga liczba) będziemy definiować następne opcje umieszczane pod poprzednią. Jeśli w nawiasie umieścimy po przecinkach jeszcze kolejne liczby, to będziemy definiować kolejne "podmenu", czyli znane chyba wszystkim "odgałęzienia" pojawiające się po wybraniu danej opcji w którymś menu. Aby zdefiniowane instrukcją "Menu$" menu pojawiło się na ekranie, użyć należy instrukcji **Menu On**.

Myślę, że wszystkie ewentualne wątpliwości związane z definiowaniem menu rozwieje prosty przykład.

```
Rem Definiowanie menu
Screen Open 0,640,256,16,Hires
Menu$(1)=" Aaaaaa "
Menu$(2)=" Bbbbbb "
Menu$(3)=" Cccec "
Menu$(1,1)=" aaaaa-1 "
Menu$(1,2)=" aaaaa-2 "
Menu$(1,3)=" aaaaa-3 "
Menu$(1,2,1)="aaaaa-2-1 "
Menu$(1,2,2)="aaaaa-2-2 "
Menu$(2,1)=" bbbbbbbb-1 "
Menu$(2,2)=" bbbbbbbb-2 "
Menu$(3,1)=" cccccccccccc-1 "
Menu On
Wait Key
End
```

Aby wykorzystać wspomniane [redacted] początku możliwości zmiany wyglądu menu trzeba zastosować specjalne komendy umieszczane w definicji menu. Zestaw takich komend oddzielonych dwukropkami i ujęty w nawiasy należy umieścić w zmiennej tekstowej opisującej tytuł menu lub opcji. Każda taka komenda składa się z dwóch liter będących najczęściej pierwszymi dwoma literami analogicznej instrukcji Basica. Można więc zapisywać pełne nazwy instrukcji, gdyż uwzględniane będą i tak tylko pierwsze dwie litery, a pozostałe zostaną zignorowane. Opisując te instrukcje będę podawać w nawiasach ich pełne nazwy, co być może ułatwi zapamiętanie.

LO x,y (Locate)

x,y - współrzędne punktu

Komenda ta ustawia kursor graficzny w punkcie o współrzędnych x,y w stosunku do lewego górnego rogu listwy menu.

BO n (Bob)

n - numer boba

Komenda ta umieszcza w pozycji kursora graficznego boba o numerze n. Tzw. "gorący punkt" nie jest przy tym brany pod uwagę, [redacted] współrzędne odnoszą się zawsze do lewego górnego rogu obrazu boba. Kolor nr 0 traktowany jest jako przezroczysty.

IC n (Icon)

n - numer ikony



Rysuje w pozycji kursora ikonę o numerze n. W przeciwieństwie do bobów umieszczanych w menu poprzednią instrukcją, kolor nr 0 nie jest traktowany jako przezroczysty.

IN n,k (Ink)

n=1 - zmiana koloru atramentu
n=2 - zmiana koloru tła
n=3 - zmiana koloru obrysu
[illegible] - numer koloru

Instrukcja ta działa analogicznie do Ink w Basicu. Pozwala [illegible] przyporządkowanie wybranego koloru (z dostępnej palety) dla atramentu, tła lub obrysu.

SF n (Sfont)

n - numer zestawu znaków

Ustala zestaw znaków (font) dla wszystkich napisów w menu. Używać można 2 zestawów z ROMu oraz po wykonaniu z poziomu Basica instrukcji Get Fonts wszystkich dostępnych na dysku.

SS n (Sstyle)

n - wartość bitowa

Pozwala ustawić atrybuty bieżącego zestawu znaków. Trzy kolejne bity określają: podkreślenie, pogrubienie, pochylenie. Nadanie wybranym bitom wartości "1" włącza dany efekt. Składnię tej instrukcji najlepiej będzie zademonstrować na przykładzie.

SS 011

Takie ustawienie bitów daje pismo pogrubione i pochylone.

LI x,y (Line)

x,y - współrzędne punktu

Rysuje linię, która łączy bieżącą pozycję kursora graficznego z punktem o podanych współrzędnych.

BA x,y (bar)

x,y - współrzędne punktu

Rysuje wypełniony prostokąt, którego lewy górny róg będzie znajdował się w bieżącym położeniu kursora graficznego, [illegible] prawy dolny w miejscu określonym współrzędnymi x i y.

OU n (Outline)

n=0 lub 1

Otacza ramką w kolorze obrysu wszystkie rysowane po wykonaniu tej instrukcji prostokąty. Komenda ta z parametrem "1" włącza rysowanie obrysu, a z parametrem "0" wyłącza.

EL a,b (Elipse)

a,b - długości promieni

Rysuje elipsę o promieniach określonych parametrami a,b i środku w bieżącym położeniu kursora. Ustawienie równych długości promieni pozwala na narysowanie okręgu.

Praktyczne zastosowanie opisanych wyżej komend można obejrzeć w przykładzie "Zmiany wyglądu menu". Dobór efektów jest raczej przypadkowy i nad estetyką takiego menu można by dyskutować, zależało mi jednak na pokazaniu zasad użycia instrukcji specjalnych, włączanych do definicji menu.

```
Rem Zmiany wyglądu menu
Screen Open 0,640,256,16,Hires
Menu$(1)=" Aaaaaa "
Menu$(2)=" Bbbbbb "
Menu$(3)=" Ccccc "
Menu$(1,1)="(IN 1,4) aaaaa-1 "
Menu$(1,2)=" aaaaa-2 "
Menu$(1,3)="(LO 0,0:IN 1,4:BA 75,20:LO 0,6:IN 2,4:IN 1,3) aaaaa-3"
Menu$(1,2,1)="(IN 2,5)aaaaa-2-1 "
Menu$(1,2,2)="aaaaa-2-2 "
Menu$(2,1)="(IN 1,5) bbbbbbbb-1"
Menu$(2,2)="(SS 1) bbbbbbbb-2(SS 0)"
Menu$(3,1)="(SS 010) cccccccccccc-1(SS 0)"
Menu On
Wait Key
End
```

Ponieważ definiowanie mamy za sobą, czas już zająć się obsługą naszego menu, czyli zasadami odczytywania wyboru dokonanego przez użytkownika. Starym zwyczajem na początek przedstawię kilka instrukcji z tym związanych.

Choice(n)

Jeśli w miejsce n wstawimy "1", to instrukcja ta będzie przyjmować wartość równą numerowi aktualnie wybranego menu. Wystarczy, że któreś menu będzie "rozwinięte", aby funkcja ta przyjęła wartość równą jego numerowi (poszczególne opcje mogą pozostać niewybrane). Jeżeli n=2 funkcja będzie przyjmowała wartość równą numerowi opcji wybranej w dowolnym menu. Nadawanie parametrowi n większych wartości (3,4,5.. itd.) spowoduje, że funkcja będzie przyjmować numery opcji wybranych w kolejnych poziomach (odgałęzieniach). Należy pamiętać, że menu numerowane są od lewej do prawej, [illegible] opcje od góry do dołu, zaczynając od "1".

On Menu Proc aa,bb,cc .....

aa,bb,cc - tytuły procedur

Instrukcja ta przyporządkowuje poszczególnym menu podane procedury. Przy czym pierwszy podany tytuł odnosi się do pierwszego menu, drugi do drugiego, itd. (licząc od lewej). Tak przypisana procedura jest wywoływana, gdy użytkownik wybierze jakąkolwiek opcję z danego menu. Z tego względu procedura taka musi rozpoczynać się od sprawdzenia, która opcja została wybrana (instrukcja Choice(n)), aby mogły być podjęte konkretne działania związane z opcją wybraną przez użytkownika. Podstawową zaletą takiego systemu obsługi menu jest to, że odpowiednia procedura jest uruchamiana dopiero po wybraniu jakiejś opcji w danym menu, [illegible] przez pozostały czas jest ona tylko "testowana" z wykorzystaniem przerwań, co nie powoduje zakłóceń w wykonywanym programie, aż do chwili wybrania opcji. Po wykonaniu tak wywołanej procedury, program powraca do instrukcji następującej po On Menu Proc.

On Menu Gosub aa,bb,cc .....

aa,bb,cc - etykiety

Instrukcja ta działa podobnie do poprzedniej, z tą różnicą, że po wybraniu jakiejś opcji program wykonuje skok do podanej etykiety zamiast uruchamiania danej procedury.

On Menu Goto aa,bb,cc .....

aa,bb,cc - etykiety

Działa tak jak poprzednia, lecz po wykonaniu podprogramu nie następuje powrót do miejsca jego wywołania (jak przy Basicowym Goto).

On Menu On/Off

Aktywuje lub dezaktywuje komendy On Menu Proc/Gosub/Goto. Aktywacja jest wymagana na początku programu wykorzystującego menu i [illegible] końcu każdej procedury obsługującej menu (wywoływanej przez wymienione wcześniej komendy). On Menu Off jest zazwyczaj używane, gdy wykonywane są długie operacje, które nie powinny być przerywane przez wybieranie innych opcji z menu np. zapis na dysk.

Menu To Bank n

n - numer banku

Instrukcja ta zapisuje w podanym banku pamięci definicję całego menu. Bank ten jest zapisywany razem [illegible] programem, co pozwala [illegible] usunięcie [illegible] programu instrukcji definiujących menu (skrócenie programu). W ich miejscu umieścić należy jedynie instrukcję odtwarzającą definicję na podstawie danych zawartych w banku, czyli:

Bank To Menu n

n - numer banku

Menu takie trzeba potem normalnie aktywować komendą Menu On.

Menu [illegible] x,y

x,y - współrzędne punktu

Instrukcja ta ustawia punkt początkowy pierwszego poziomu menu, w miejscu określonym współrzędnymi x i y. Pozycje wszystkich pozostałych opcji obliczane są właśnie w stosunku do tego punktu, więc jego przemieszczenie jest równoznaczne z przesunięciem całego menu.

Set Menu(a,b,c ...) To x,y

a,b,c - definicja opcji

x,y - współrzędne punktu

Komenda ta ustawia lewy górny róg wybranej, pojedynczej opcji w punkcie o współrzędnych x,y w stosunku do poprzedniego

poziomu. Definicja opcji jest taka jak przy instrukcji Menu$(a,b,c ...). Użycie tej instrukcji przemieszcza także wszystkie poziomy poniżej tego, w którym znajdowała się przemieszczana opcja.

Menu Mouse On/Off

Standardowo menu i jego poszczególne opcje można przesuwać po ekranie za pomocą myszki, poprzez przytrzymanie na wybranej opcji wciśniętego lewego klawisza myszki i przesuwanie jej po stole. Instrukcja ta powoduje, że menu wyświetlane będzie po wciśnięciu prawego klawisza myszki w miejscu aktualnego położenia wskaźnika. Oczywiście On włącza, a Off wyłącza tę funkcję.

Menu Key(a,b,c ...) To a$

a,b,c - numery pozycji menu
a$ - jednoznakowa zmienna tekstowa

Instrukcja ta pozwala na przypisanie dowolnej pozycji menu jednego z klawiszy. Definicja opcji jest identyczna jak przy instrukcji "Menu$", a zmienna tekstowa powinna zawierać znak odpowiadający wybranemu klawiszowi. Jeśli znaków będzie więcej, wykorzystany zostanie pierwszy, a pozostałe zostaną zignorowane. Wciśnięcie tak zdefiniowanego klawisza wywoła efekt taki sam jak wybranie opcji myszką.

Menu Key(a,b,c ...) To k,n

a,b,c - numery pozycji menu
k - kod wybranego klawisza
n - wartość bitowa określająca klawisze kontrolne

Jest to w zasadzie modyfikacja poprzedniej komendy polegająca na zmianie definicji klawisza uruchamiającego podaną opcję. Zmiennej k można nadawać wartości odpowiadające kodom wybranych klawiszy. Np. klawisze funkcyjne mają kody od 80 do 89 (F1 do F10), Help - 95, a Esc - 69. Zmienna n natomiast, jest wartością bitową, której kolejne bity odpowiadają klawiszom kontrolnym.

Bit	Klawisz
0	lewy Shift
1	prawy Shift
2	Caps Lock
3	CTRL
4	lewy Alt
1	prawy Alt
6	lewa Amiga
7	prawa Amiga

Nadając odpowiednim bitom wartość "1" możemy ustalić, który z klawiszy kontrolnych musi być wciśnięty razem z klawiszem o podanym kodzie, aby uruchomić wybraną opcję. Nadając wartość "1" kilku bitom można tworzyć kombinacje klawiszy np. CTRL+Alt+D. Podstawową zaletą tej instrukcji jest możliwość wykorzystania klawiszy nie mających przyporządkowanych żadnych znaków.

Teraz chciałbym przedstawić przykład praktycznego zastosowania opisanych instrukcji. Programik "Odczytywanie menu" definiuje proste menu i pozwala na uruchamianie procedur przez wybranie odpowiedniej opcji. Procedury te zawierają jedynie instrukcję powodującą wyświetlenie komunikatu o ich uruchomieniu, gdyż jest to przykład. W konkretnym programie, powinny w nich znaleźć się komendy wykonujące zadane przez użytkownika (przez wybranie opcji) operacje. Program główny również wykonuje czysto "przykładową" operację, ale na tak wykonanym "pulsującym" napisie można dobrze zaobserwować zatrzymanie programu głównego na czas wykonywania wyboru opcji w menu.

```
Rem Odczytywanie menu
Screen Open 0,640,256,16,Hires
Curs Off
Menu$(1)=" Aaaaaa "
Menu$(2)=" Bbbbbb "
Menu$(3)=" Ccccc  "
Menu$(1,1)=" aaaaa-1  "
Menu$(1,2)=" aaaaa-2  "
Menu$(1,3)=" aaaaa-3  "
Menu$(1,2,1)="aaaaa-2-1 "
Menu$(1,2,2)="aaaaa-2-2 "
Menu$(2,1)=" bbbbbbbb-1  "
Menu$(2,2)=" bbbbbbbb-2  "
Menu$(3,1)=" cccccccccccc-1 "
Menu On
On Menu Proc MENU_A,MENU_B,MENU_C
On Menu On

Do
Locate ,10
Centre "Wybierz opcję z menu !!"
Wait 20
Locate ,10
Centre "                       "
Wait 20
Loop

Procedure MENU_A
If Choice(2)=1 Then A_1
If Choice(3)=1 Then A_2_1
If Choice(3)=2 Then A_2_2
If Choice(2)=3 Then A_3
On Menu On
End Proc
Procedure MENU_B
If Choice(2)=1 Then B_1
If Choice(2)=2 Then B_2
On Menu On
End Proc
Procedure MENU_C
If Choice(2)=1 Then C_1
On Menu On
End Proc

Procedure A_1
Home
Print "Uruchomileś procedurę A_1  "
End Proc
Procedure A_2_1
Home
Print "Uruchomileś procedurę A_2_1"
End Proc
Procedure A_2_2
Home
Print "Uruchomileś procedurę A_2_2"
End Proc
Procedure A_3
Home
Print "Uruchomileś procedurę A_3  "
End Proc
Procedure B_1
Home
Print "Uruchomileś procedurę B_1  "
End Proc
Procedure B_2
Home
Print "Uruchomileś procedurę B_2  "
End Proc
Procedure C_1
Home
Print "Uruchomileś procedurę C_1  "
End Proc
End
```

Myślę, że przeanalizowanie tego programiku pomoże wszystkim w wyposażeniu swoich programów w typowe menu rozwijane z górnej belki. Jak łatwo zauważyć, jego zdefiniowanie jest dosyć proste, a późniejsza obsługa bardzo wygodna. Z tego też względu menu warto stosować wszędzie tam, gdzie istnieje potrzeba ingerowania użytkownika w pracę programu i dokonywania wyboru wśród wielu funkcji. Na tym kończę i jak zwykle zachęcam do lektury kolejnego odcinka naszego cyklu poświęconego różnym "efektom specjalnym". Mam tu na myśli metody zmiany zawartości ekranu, czyli efekty zbliżone do znanych z programu SCALA oraz różnego rodzaju tęcze i rotacje barw. ❑



AREXX cz.5

Izabela Skibińska

Dwa podstawowe zagadnienia, którymi zajmiemy się [illegible] ostatnim odcinku naszego kursu, to komunikacja programów [illegible] pośrednictwem Arexxa oraz korzystanie z bardzo rozbudowanego mechanizmu śledzenia programów.

Wymiana danych

Jako przykład do zobrazowania pierwszego problemu posłuży nam omawiany już wcześniej "Art Department Proffesional" oraz bardzo interesujący program (rozpowszechniany przez tę samą firmę) Mand2000 służący do generacji fraktali. W przykładowym skrypcie najpierw generujemy programem Mand2000 [illegible] różnych fragmentów zbioru Mandelbrota, [illegible] następnie za pośrednictwem ADPro układamy je w jedną kompozycję i podpisujemy. Aby zmieścić wszystkie obrazki na jednym definiujemy odpowiednie wymiary okna do generacji fraktali (komenda SIZEWINDOW). Wynik działania skryptu zostanie umieszczony na ekranie o wymiarach 320x512 i 32 kolorach (patrz rysunek). Ważne jest więc ustawienie takich samych parametrów pod Mand2000 (szczególnie liczby kolorów), niestety nie mam pojęcia jak zdefiniować te wartości z poziomu Arexxa dla Mand2000 i co za tym idzie, jeśli również nie wiecie, należy ustawić je "ręcznie". Komenda GOTOGOOD n zapoczątkuje generację n-tego fragmentu zbioru Mandelbrota. Kolejność określona jest w podmenu "Good spots". Na uwagę zasługuje wywołanie rekurencyjnej procedury czekaj, której zadaniem jest sprawdzanie czy operacja została zakończona. Niezwykle istotną sprawą jest tu konieczność ustawienia priorytetu samego programu na 0. W przypadku, gdy wartość ta jest mniejsza, wykonująca się jednocześnie rekurencyjna procedura spowoduje szybkie zapełnienie stosu i w rezultacie wkrótce otrzymamy komunikat "Insufficient memory".

W tym miejscu mała dygresja. Gdy przedstawialiśmy prosty wariant realizacji programu z portem Arexxa w części trzeciej kursu, nie uwzględnialiśmy możliwości wykonywania pewnych rozkazów programu asynchronicznie tzn. w ten sposób, aby główny program nie był zajęty jedynie wykonaniem danej komendy Arexxa. W Mand2000 najwyraźniej komenda GETATRR pobierająca różne informacje o programie (w naszym przypadku status wykonania zadania) jest właśnie asynchroniczna. Aby zrealizować taki wariant, oczywiście należałoby stworzyć oddzielny proces, ale jest to już temat na osobny (obszerny zresztą) artykuł.

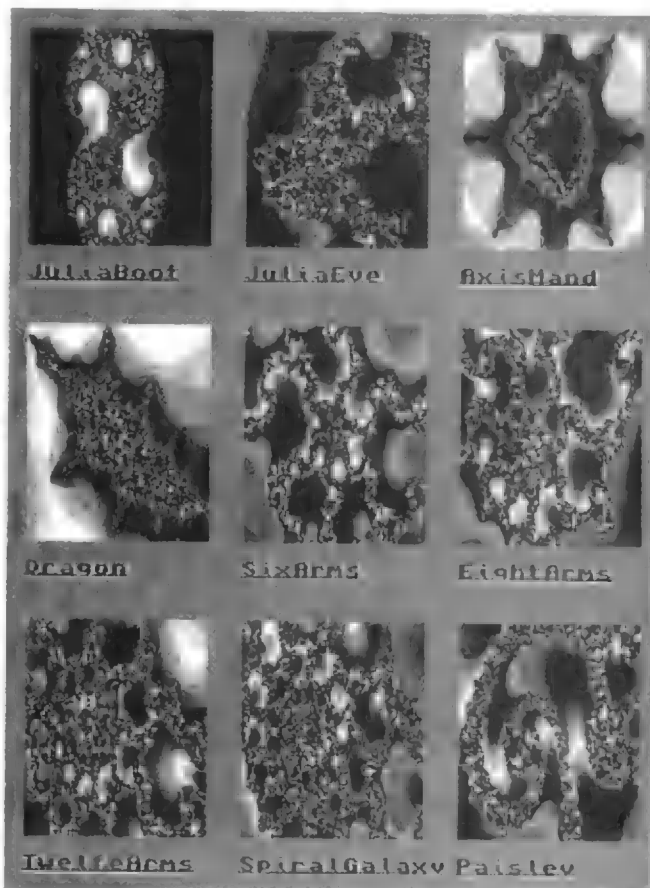
Wracając do przykładu, gdy Mand2000 zrealizuje swoje zadanie, "pałeczka" przekazywana jest do ADPRO, który czyta wygenerowane obrazki i ustawia je na szarym tle. Szczegóły przedstawiają się następująco.

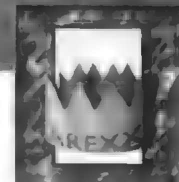
Ustawiamy format ładowanej grafiki na "BACKDROP", aby spowodować generację tła, którego atrybuty określone są przez argumenty komendy LOAD. Tak więc tworzymy 24-bitowe tło o rozmiarach 320x512 i jednolitym szarym kolorze (150 150 150), [illegible] które wkomponujemy obrazki fraktali.

ADPro umożliwia nam dwojaki sposób wczytywania grafiki. Albo zastępujemy dane w buforze nowo wczytywanymi (ustawienie REPLC w panelu głównym programu) albo miksujemy dane już istniejące z nowymi (COMP) w odpowiadający nam sposób. Z poziomu Arexxa sposób mieszania danych możemy określić podając odpowiednie opcje [illegible] rozkazu LOAD. Pełna definiacja tej komendy jest postaci:

LOAD FILENAME [L_OPTS] [C_OPTS] [FORCE PALETTE]

gdzie L_OPTS są to opcje zależne od formatu grafiki, natomiast C_OPTS definiują nam sposób mieszania danych.





Opcje te mogą przyjmować dwie formy:

1) LEFT_EDGE TOP_EDGE MIX_LEVEL

2) LEFT_EDGE TOP_EDGE MIX_LEVEL RT GT BT

W obu wersjach należy określić pozycje nowego obrazka względem lewego górnego rogu starej grafiki oraz stopień wymieszania (od 1 do 100). Druga forma pozwala nam na określenie koloru przezroczystego. Dla porządku wspomnę, że trzecia opcja rozkazu LOAD pozwala na zastąpienie aktualnej palety kolorów tymi z obrazka, nawet gdy paleta jest "zamrożona" (LOCKED).

Wracając do naszego przykładu, to aby "przykryć" tło wczytywanymi fraktalami, należy ustawić poziom zmieszania na 100. Operator "TEXT_VISUAL" umożliwi nam wkomponowanie tekstu w grafikę. Posiada on wiele argumentów określanych przy pomocy słów kluczowych, których dużą część wykorzystaliśmy w naszym skrypcie. Niejasne mogą być tu słowa kluczowe SET_BLUR oraz EMBOSS_DIRECTION. Pierwsze z nich określa wkomponowanie znaku w podłoże. Wartość -1 określa całkowity brak anti-aliasingu, dzięki temu litery będą bardziej wyraziste (0 - wartość maksymalna , 16 - minimalna). Drugie słowo kluczowe zapewni nadanie trójwymiarowego wyglądu znakom. Parametr OUT powoduje, że litera jest wystająca (w odróżnieniu od IN dającego wrażenie wklęsłości). Stopień nasilenia wymienionej wyżej cechy regulowany jest parametrem po "SET_EMBOSS".

Ważnymi komendami dla samego już ADPRO są RENDER_TYPE oraz SCREEN_TYPE. Pierwsza z nich określa w co mają być zamienione "surowe" 24-bitowe dane, po wykonaniu rozkazu EXECUTE. Rozkaz ten akceptuje jeden z następujących argumentów: 2, 4, 8, 16, 32, 64, 128, 256, EHB, HAM, HAM8 lub CUST. Rozkaz SCREEN_TYPE określa naturalnie rodzaj ekranu i przyjmuje dowolną kombinacje parametrów (sumę) określonych liczbami:

HIRES = 1
INTERLACE = 2
PAL =4
HOVERSCAN = [illegible]
VOVERSCAN = 16

W skrypcie użyliśmy też komendy ADPRO_DISPLAY mającej identyczne działanie jak przy kliknięciu na REDISPLAY w panelu głównym programu.

Na tym skończymy zabawę z ADPRO i zajmiemy się już samym Arexxem.

```
/* mandadpro.rexx */
SPOT.1 = "MANDELBROT"
SPOT.2 = "JuliaBoot"
SPOT.3 = "JuliaEye"
SPOT.4 = "AxisMand"
SPOT.5 = "Dragon"
SPOT.6 = "SixArms"
SPOT.7 = "EightArms"
SPOT.8 = "TwelfeArms"
SPOT.9 = "SpiralGalaxy"
SPOT.10  = "Paisley"

  ADDRESS'MAND2000.1"
  SIZEWINDOW 84 110


do i=2 to 10

  GOTOGOOD
  call czekaj
  SAVEAS 'NAME = ram:"spot.i 1

end
CLOSE FORCE

ADDRESS "ADPro"
ADPRO_TO_FRONT
RENDER_TYPE 32
SCREEN_TYPE 6 /* PAL + INTERLACE */


LFORMAT "BACKDROP"
LOAD "XXX" 320 512 "COLOR" 150 150 150

ARGUMENTS = ' SET_COLORS 0 0 0 SET_BLUR "-1" SET_TEXT_
        STYLE 1 SET_FONT_SIZE 8 FONT_NAME "topaz" EMBOSS
                          DIRECTION OUT SET_EMBOSS 100'
LFORMAT 'IFF'

  do i=2 to 4
    LOAD "ram:"spot.i (i-2)*100+17 10 100
    OPERATOR "TEXT_VISUAL" ARGUMENTS SET
       XOFFSET (i-2)*100+16 SET_YOFFSET 130 STRING spot.i ,
    DRAW
    end

  do i=5 to 7

    LOAD  "ram:"spot.i (i-5)*100+17 160 100
    OPERATOR "TEXT_VISUAL" ARGUMENTS SET_XOFFSET
              (i-5)*100+16 SET_YOFFSET 280 STRING spot.i ,
    DRAW

end

  do i=8 to 10
  LOAD  "ram:"spot.i (i-8)*100+17 310 100
  OPERATOR "TEXT_VISUAL" ARGUMENTS SET_XOFFSET
              (i-8)*100+16 SET_YOFFSET 430 STRING spot.i ,
    DRAW
  end

EXECUTE
ADPRO_DISPLAY
exit

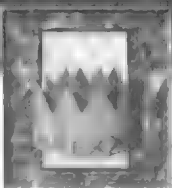

czekaj: procedure
  GETATTR stem PROJ
  if (PROJ.DONE = 0) then
  call czekaj
  return
```

Uwaga! Tekst wytłuszczony należy wpisywać w jednej linii!

Skoro już jesteśmy przy tak ważnym zagadnieniu jak wymiana danych między programami warto wspomnieć o bardzo ciekawym mechanizmie komunikacji w Arexxie jaki zapewnia nam tzw. ClipLista. Wyobraźmy sobie, że chcemy uniezależnić od siebie czasowo część pracy wykonywaną przez program Mand2000 i ADPRO. Musimy zatem zdefiniować pewne zmienne globalne zawierające takie parametry jak ilość obrazków, ich nazwy. Do tego celu służy właśnie wspomniana ClipLista. Stanowi ona ciąg par postaci ("Name",val), gdzie interesująca nas wartość val jest odnajdywana po swej nazwie "Name". Do manipulacji zawartością takiej listy służą dwie funkcje SetClip() i GetClip(). Poniższy przykład przedstawia działanie tych dwóch funkcji. Jego zadaniem będzie odpowiednie skomunikowanie programów ADPRO i Mand2000, tak aby mogły wykonać to samo zadanie co poprzednio tylko, że niezależnie (nie w jednym skrypcie).

```
/* fragment skryptu od strony Mand2000 */
files=""
do i=2 to 10
     files = files "ram:"spot.i
end
setclip("Pliki",files)
setclip("Od",2)

/* od strony ADPro */
     pliki = getclip("Pliki")
     j = getclip("Od")
     if pliki~=""  then  do
        ile = words(pliki)
```



```
    do i=j to ile
      spot.i = word(pliki,i)
    end
  end
  else exit
```

Bardzo duże możliwości daje mechanizm oferowany przez ClipListę w połączeniu z rozkazem Arexxa INTERPRET. Wtedy możemy taką listę traktować niemalże jak plik prototypowy w C.

```
defs=' pi = 3.1415;e = 2.718"
setclip("constans",defs)
cons = getclip("constans")
interpret cons
say pi
say e
```

Jak widać komenda INTERPRET powoduje zinterpretowanie danego ciągu znakowego jako bloku instrukcji arexxowych. Należy zwrócić uwagę na konieczność oddzielania instrukcji średnikiem, gdyż jest to wymagane dla pisania rozkazów w jednej linii (o czym zresztą już wspominaliśmy).

Zmienne złożone

W skrypcie mandadpro.rexx korzystaliśmy z tzw. zmiennych złożonych. Zmienne te tak jak zwykłe nie wymagają wcześniejszych deklaracji ani rezerwacji pamięci. Podobnie jak w innych językach programowania tablice czy struktury, tak też w Arexxie zmienne złożone umożliwiają manipulację uporządkowanymi danymi. Bardzo wygodną sprawą jest to, że zmienna złożona może zawierać elementy różnych typów. Zmienną taką tworzy się w Arexxie przez podanie dowolej ilości symboli (liczb bądź ciągów znakowych) oddzielonych znakiem kropki. Twór ten traktowany jest przez interpreter podobnie jak zwykła zmienna. Tak więc, gdy nie jest zainicjalizowana, przyjmuje swoją nazwę za wartość, rozkazem DROP możemy natomiast, [redacted] zmiennej posiadającej wartość uczynić ponownie niezainicjalizowaną. Króciutki przykład:.

```
/*  */
say array.field1.field2.field3
    do i=1 to 5
        do j=1 to 5
            do k=1 to 2
                array.i.j.k = random(1,100)
            end
        end
    end
    field1 = 4;field2=2;field3=1;field4=10000
    say array.field1.field2.field3
    array.= 101
    say array.field1.field2.field3
    say array.field1.field2.field3.field4
    say  array.field1
```

W powyższym skrypcie tworzymy trójwymiarową tablicę. Na uwagę zasługuje instrukcja "array.=101", która inicjalizuje nam całą tablicę tą warością i to niezależnie do którego poziomu się odwołamy. Tak więc zmienne złożone mogą powstawać nie tylko w momencie przypisania im określonej wartości, ale też w chwili odwołania się do nich!

"Odpluskwianie"

Jak już wspominaliśmy, możliwości Arexxa w zakresie śledzenia programów są bardzo duże. Kluczową rolę w całym procesie pełnią rozkazy SIGNAL i TRACE. Pewnie większość z Was zauważyła, że wykonujący się skrypt arexxowy można przerwać kombinacją klawiszy CTRL-C. Arexx posiada możliwość przejęcia tego typu sygnałów i odpowiedniego zareagowania na nie. Do tego służy właśnie rozkaz SIGNAL. Jego składnia jest następująca:

SIGNAL {ON|OF} war

lub

SIGNAL label

gdzie war jest parametrem określającym rodzaj zdarzenia. Są to:

BREAK_C - wciśnięcie Ctrl-C
BREAK_D - wciśnięcie Ctrl-D
BREAK_F - wciśnięcie Ctrl-E
BREAK_F - wciśnięcie Ctrl-F
ERROR - błąd pochodzący z komend zewnętrznego "hostu" (zmienna RC różna od zera)
HALT - wywołanie zewnętrznej komendy HI, zatrzymującej wszystkie skrypty Arexxa
IOERR - błąd wejścia/wyjścia
NOVALUE - zmienna nie zainicjalizowana
SYNTAX - błąd składniowy

Parametry te pełnią też rolę etykiet, do których zostanie wykonany skok w przypadku zajścia określonej przez nie sytuacji. Słowami kluczowymi ON i OFF możemy włączyć lub wyłączyć reakcję na wyżej wymienione zdarzenia.

Druga forma rozkazu SIGNAL powoduje bezwarunkowy skok do etykiety określonej wyrażeniem "label". Oto przykład:

```
/* signals.rexx */
SIGNAL ON HALT
SIGNAL ON BREAK_C
SIGNAL ON BREAK_D
SIGNAL ON BREAK_F
SIGNAL ON IOERR
SIGNAL ON NOVALUE
SIGNAL ON SYNTAX

begin:

  say "Press Ctrl-C  or run HI"
  do forever
  end

exit

HALT:
say "Zatrzymanie przez program  HI"
SIGNAL begin

BREAK_C:
say "Zwyczajne Ctrl-C"
SIGNAL ON BREAK_C
SIGNAL begin

BREAK_F:
  say "Ctrl-F "
  say k

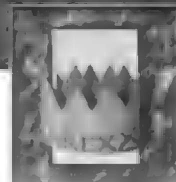
BREAK_D:
  say "Ctrl-D"
  IF 1 THEN SELECT

NOVALUE:
  say "Niezainicjalizowana zmienna"
  SIGNAL ON NOVALUE
  SIGNAL ON BREAK_F
  SIGNAL begin

SYNTAX:
  say "Blad skladniowy"
SIGNAL ON SYNTAX
SIGNAL ON BREAK_D
  SIGNAL begin
```

W skrypcie signals.rexx na uwagę zasługuje konieczność ponownego ustawienia sygnału, gdy już jeden tego typu został przyjęty. Dlatego też opuszczenie programu jest możliwe przez dwukrotne wywołanie programu HI.

```
/* */
SIGNAL ON ERROR
```



```
ADDRESS "ADPro"
EXECUT
exit
ERROR:
  say "Zły rozkaz"
return
```

Powyższy przykład ilustruje możliwość przechwycenia złej komendy dla "hostu".

Kolejna instrukcja TRACE pozwala nam na prześledzenie w programie odpowiednich symboli. Posiada dużo parametrów, my opiszemy tylko najważniejsze z nich:

ALL - śledzi wszystkie symbole
COMMANDS - śledzi komendy dla zewnętrznego programu zanim zostaną tam przesłane
INTERMEDIATES - pokazuje wszystkie wartości pośrednie przy wartościowaniu zmiennych, a także śledzi wszystkie symbole
LABELS - śledzi wszystkie etykiety
RESULTS - wszystkie instrukcje i funkcje zostaną wylistowane przed wywołaniem, po czym zostanie podany ich wynik
SCAN - zostaje sprawdzona poprawność programu, ale prgram nie jest uruchomiony
? - śledzenie interakcyjne (możliwość modyfikacji zmiennych)
! - nie będą przesyłane komendy do zewnętrznych "hostów" w celu uniknięcia nieporządanych skutków
OFF - wyłączenie jakiegokolwiek śledzenia
Przykład:

```
/*            */
TRACE INTERMEDIATES

arr.1 =  "filename1"
arr.2 =  "filename2"
        files= ""
        do i=1 to 2
          files = files "ram:"arr.i
        end
k=words(files)
```

Przy wykonaniu powyższego skryptu otrzymaliśmy, dzięki odpowiednim prefiksom, informację o rodzaju wszystkich występujących w nim zmiennych symboli instrukcji łącznie z ich wartościami pośrednimi i wynikami:

```
 5 *-* arr.1 = "filename1";
     >L> "filename1"
     >>> "filename1"
     >C> "ARR.1"
  6 *-* arr.2 = "filename2";
     >L> "filename2"
     >>> "filename2"
     >C> "ARR.2"
  7 *-* ;
  8 *-* files="";
     >L> ""
     >>> ""
  9 *-* do i=1 to 2;
     >L> "1"
     >>> "1"
     >L> "2"
      >V> "1"
  10 *-* files = files "ram:"arr.i;
      >V> ""
      >L> "ram:"
      >C> "ARR.1"
      >V> "filename1"
      >O> " ram:"
      >O> " ram:filename1"
      >>> " ram:filename1"
 11 *-* end;
  9 *-* do i=1 to 2;
      >V> "1"
      >O> "2"
      >>> "2"
 10 *-* files = files "ram:"arr.i;
      >V> " ram:filename1"
      >L> "ram:"
      >C> "ARR.2"
      >V> "filename2"
      >O> " ram:filename1 ram:"
      >O> " ram:filename1 ram:filename2"
      >>> " ram:filename1 ram:filename2"
 11 *-* end;
  9 *-* do i=1 to 2;
      >V> "2"
      >O> "3"
      >>> "3"
12 *-* k=words(files);
     >V> " ram:filename1 ram:filename2"
     >F> "2"
     >>> "2"
```

Jak można wywnioskować z otrzymanego listingu tzw. kody prefiksowe śledzenia oznaczają:

>L> - etykieta (też dowolny napis)
>>> - rezultat wyrażenia
>C> - symbol złożony
>V> - wartość zmiennej
>O> - wynik operacji dwuparametrowej
>F> - rezultat wywołanej funkcji

W śledzeniu interakcyjnym przy każdej instrukcji mamy możliwość wprowadzenie dowolnego wyrażenia, które następnie będzie interpretowane przez program. Prompt dla użytkownika jest oznaczany prefiksem śledzenia - ">+>". Ciekawe możliwości daje połączenie rozkazów TRACE COMMANDS i TRACE! Można wtedy prześledzić jakie komendy zostaną wysłane do "hostów" bez ich wykonania.

Zachęcam gorąco do eksperymentów z rozkazami TRACE i SIGNAL. Skoro ostatnie akapity naszego kursu o Arexxie poświęciliśmy tropieniu błędów, winna jestem parę sprostowań. W długim procesie przygotowawczym do druku w artykułach o Arexxie pojawiły się drobne przekłamania, zmieniające pierwotny sens wypowiedzi. Tak też w części pierwszej czytamy (1 kolumna 6 akapit):

"zachodzi odpowiedni "dialog" za pośrednictwem specjalnej message, wzbogaconej w stosunku do zwykłej struktury message o dodatkowe informacje pochodzące z exec.library",

podczas, gdy powinno być:

"... zachodzi odpowiedni "dialog" za pośrednictwem specjalnej message wzbogaconej o dodatkowe informacje w stosunku do zwykłej struktury message z exec.library "

W części trzeciej natomiast jest napisane (koniec drugiej kolumny):

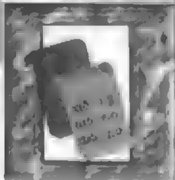
"Istotnymi wartościami dla struktury RexxMsg w momencie jej tworzenia jest wskaźnik i nazwa portu, gdzie ma być odsyłana po wywołaniu przez odbiorcę funkcji odpowiedź (ReplyMsg).

zamiast:

"..., gdzie ma być odsyłana po wywołaniu przez odbiorcę funkcji ReplyMsg."

Ponadto część komentarza do Listingu 2 powinna brzmieć:

/* gdzie command np. "checkdrive\df0:" (składnia komend Arexxa Virus */ /* Checkera wymaga, aby argumenty były odzielone od komendy znakiem " */ (W C dwa znaki backslash oznaczają dopiero jeden) ❑



AMIGA DOS cz. 1 jeszcze raz

Wojciech [illegible]

Większość średnio zaawansowanych użytkowników Amigi doskonale wie, że DOS to system zarządzania plikami. Niektórzy czarodzieje wiedzą nawet na czym polega hierarchia zbiorów oraz jak ją pokazać zaklęciem DIR. Na tym jednak AmigaDOS się nie kończy, jest to raczej wstęp do jego możliwości. Jak korzystać z DOSa w sposób bardziej zaawansowany możesz przeczytać w poniższym tekście.

Zanim przejdziemy do dokładniejszego opisu kilka drobnych uwag. Amiga posiada system bibliotek zawierających funkcje pomagające w budowaniu programów. Najbardziej "zagnieżdżona" jest Exec.library, która zarządza krytyczną częścią komputera czyli procesorem, pamięcią, urządzeniami zewnętrznymi, komunikacją pomiędzy zadaniami, itd. Kolejne biblioteki korzystają z funkcji zapewnionych przez Exec, oraz obsługują następne biblioteki. Taka hierarchiczna budowa ma bardzo dużą przewagę nad systemami "płaskimi" (np. prymitywny MS-DOS), gdyż zmiana w bibliotece bliższej "środka" pociąga za sobą automatyczne rozszerzenia w wyższych bibliotekach. Jeszcze jeden "drobiazg" - funkcje zapewniane przez biblioteki mogą być dowolnie wykorzystywane przez programy w dowolnych językach - od Assemblera poprzez C, Pascal i Arexx, aż do Basica.

Drugą z zagnieżdżonych bibliotek jest DOS.library. Obsługuje ona wszelkie operacje związane z przechowywaniem danych. DOS.library zawiera ponad sto funkcji, które pomagają w korzystaniu z dysków, przy tworzeniu nowych urządzeń logicznych, a przede wszystkim obsługiwaniu hierarchicznej struktury większości urządzeń DOSu.

Jednak użytkownik przed komputerem chce ładować program lub zmieniać katalog bez konieczności pisania programu, który będzie wykonywał owe czynności. Specjalnie w tym celu Commodore napisał szereg prostych programików mieszczących się w katalogu C:. Wpisanie komendy:

1.Work:> DIR RAM:

spowoduje wczytanie programu DIR z katalogu C, podanie mu argumentu "RAM:". Uruchomiony programik skorzysta z szeregu funkcji biblioteki DOS, aby przeczytać katalog, a nastepnie wyświetli go na ekranie. Podobny przebieg mają pozostałe komendy. Chcę abyś drogi Czytelniku zauważył dwa wnioski:

- Praktycznie cały "DOS" jest biblioteką i znajduje się wewnątrz systemu operacyjnego naszej Amigi (jest to ROM). Programy zawarte na dysku Workbench jedynie ułatwiają korzystanie z wbudowanego DOSu.
- DOS jest jedynie częścią systemu operacyjnego Amigi, nie ma on tak dużego znaczenia jak w komputerach prostszych np. IBM PC. Z tego względu zwyczajowe powiedzenie "System Amiga DOS" jest niepoprawne i wprowadza w błąd. Niektóre programy (np. zawarte na prawie każdej karcie dołączanej do komputera) w ogóle z DOSu nie korzystają, nie są nawet przez niego ładowane!

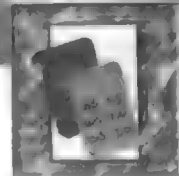
Flagi protekcji.

Każdy katalog oraz plik posiada zestaw znaczników opisujących dozwolone operacje na takim pliku. Flagi te zostały wprowadzone prawie na początku kariery Amigi, ale dopiero system 2.0 bezpośrednio je interpretował. Wcześniej były one jedynie wskazówką dla programów. Oto najważniejsze znaczniki:

R - ang. Readable - możliwy do odczytu.
W - ang. Writable - możliwy do zapisu.
D - ang. Deleteable - możliwy do kasowania.

Znaczniki znajdują się jedynie w dwóch stanach - aktywny (Set) oraz zgaszony (Clear). Znaczniki są niezależne od siebie, plik może być zapisywalny, ale bez możliwości odczytu (chyba niezbyt przydatne), odczytywalny, ale bez możliwości zapisu, itd. To samo dotyczy znacznika kasowania (D), który pozwala na użycie komendy DELETE. Zapis do pliku jest po prostu użyciem opisu pliku (ścieżka dostepu, nazwa itd.) do przechowywania innych danych. Skasowanie jest usunięciem całego obiektu. Wykonajmy pewien eksperyment:

```
Copy c:dir to Ram:
List Ram:
Protect ram:dir w
List Ram:
Copy Ram:dir to Ram:plik
Protect ram:dir r
Copy Ram:dir to Ram:plik
Protect ram:dir w
Delete ram:dir
Protect ram:dir d
Delete ram:dir
```



Czas chyba opisać składnię poleceń. Rzadko używana komenda LIST podaje nam dokładniejsze dane o bieżącym lub wybranym katalogu. Oprócz jego struktury podaje czas utworzenia, długość plików oraz stan znaczników. Rozkaz PROTECT pozwala na zmianę stanu znaczników. Uruchomiony jedynie z nazwą pliku kasuje wszystkie znaczniki. Pozwala to na zabezpieczenie pliku przed wszelkimi nieszczęściami. Podanie symbolu znacznika powoduje jego ustawienie, przy czym możesz podać kilka znaczników jednocześnie bez spacji między nimi np. RWD. Możemy również decydować o kasowaniu/ustawianiu znaczników argumentami SUB - kasuj (ang. odejmuj), ADD - ustawiaj (ang. dodaj). Ostatni parametr QUIET ma sens jedynie przy wykorzystywaniu dżokerów - wyjaśnię to nieco później.

Pozostałe znaczniki: (polskie słowo jest raczej objaśnieniem, a nie tłumaczeniem angielskiej nazwy)

E - wykonywalny (ang. Executable).
A - bezpieczny (ang. Archive).
H - schowany (ang. Hidden).
P - rezydentny (ang. Pure).
S - plik wsadowy (ang. Script).

Jak wiesz wszystkie pliki mogą być zarówno danymi jak i programami (przeważnie przetwarzającymi dane). AmigaDOS odróżnia oba typy według wewnętrznej struktury, ale dodatkowo flaga E pomaga w odróżnianiu takich plików. Ustawienie tej flagi nie umożliwi "uruchomienia" danych, ale jej skasowanie nie pozwoli na pracę prawdziwego programu.

Znacznik H pozwala na ukrycie pliku bądź katalogu przed ... nawet Commodore jeszcze się nie zdecydowało, gdyż flaga ta "chwilowo" nie działa. Jak dotąd, to nie flaga ukrywa pliki, ale Commodore flagę, gdyż nie ma o niej mowy w instrukcjach obsługi, ani nawet w opisie komendy wywoływanej jako "PROTECT ?".

Znacznik A pozwala na łatwą implementację programów tworzących kopię bezpieczeństwa. Programy takie przeważnie go ustawiają, zaznaczając, że jest on bezpieczny. Jakakolwiek modyfikacja pliku z powrotem gasi go, wskazując na konieczność utworzenia jego kopii podczas kolejnego "zrzucania" zbiorów np. na dyskietki. Znacznik S wskazuje na konieczność użycia komendy EXECUTE podczas "wykonywania" pliku wsadowego, natomiast P na możliwość użycia komendy RESIDENT.

Zwracam uwagę na pewną kwestię, która jest przemilczana w większości książeczek o Amiga DOSie. Komenda PROTECT pozwala na zmianę stanu znaczników pliku, ale tylko wtedy, gdy WSZYSTKIE poprzedzające go katalogi mają ustawioną flagę W. Przykładowo:

```
Makedir  RAM:Programy
Makedir  RAM:Programy/Komendy
Copy     c:dir to RAM:Programy/Komendy
Protect  RAM:Programy w sub
List     RAM:Programy/Komendy
Protect  RAM:Programy/Komendy/dir , skasuj flagi
List     RAM:Programy/Komendy
```

komenda PROTECT nie zmieniła stanu znaczników, gdyż katalog Programy nie umożliwiał zapisu. Dopiero gdy Programy ORAZ Komendy będą miały ustawione flagi protekcji - dopiero wtedy można będzie zmieniać stan znaczników pliku <dir>. Zwróć uwagę, że komenda Protect nie wykaże błędu!

Programy rezydentne.

Na starszych komputerach (np. IBM PC z płytą 286) programy rezydentne były czymś wyjątkowym i przeważnie tworzyły możliwość ograniczonego wykorzystywania więcej niż dwóch komend (jednak nie programów, gdyż były to raczej procedury "podwieszane" pod przerwania). W ten sposób dawano możliwość jednoczesnego drukowania (IBM - PRINT) i wykonywania innych poleceń - zaiste niesamowite!

Amiga dysponując systemem wielozadaniowym zawsze pozwala na procesy w tle, stąd "program rezydentny" ma na niej nieco inne znaczenie. Programy takie pozostają w pamięci komputera, jednak zapewniają znacznie ciekawszą możliwość jednoczesnego uruchamiania - np. z wielu okienek Shell. W ten sposób JEDNA kopia programu jest jednocześnie uruchamiana, nie tracąc dodatkowej pamięci na powielanie tego samego kodu. Zwróć uwagę, że programy takie muszą być napisane w specjalny sposób, gdyż fizycznie ten sam kod musi jednocześnie przetwarzać różne dane. Stąd też istnieje flaga protekcji P, która pozwala na użycie komendy RESIDENT wobec tego programu. Ustawienie tej flagi jest bardzo niebezpieczne, gdyż program może nie być przygotowany do pracowania jako moduł rezydentny. Standardowy Shell jak i Cli nie odróżniają struktury zwykłych programów od programów rezydentnych, robią to jedynie na podstawie flagi P. Inne programy mogą ją jednak zauważać.

Większość komend z katalogu C: pozwala na użycie komendy RESIDENT, często jest to również możliwe wobec innych "dobrze" napisanych programów. Wystarczy wpisać:
Protect <Program> P Add
Resident <Program>
a następnie próbować JEDNOCZESNIE uruchamiać testowany program, np. z różnych okienek. Jeśli nic nie wybuchnie, to możemy pozostawić flagę P, chociaż nie ma 100% pewności.

System 2.0 ma szereg komend rezydentnych już w ROMie komputera, co pozwala na ich wykorzystanie nawet bez wcześniejszego ładowania. Stworzenie pliku w katalogu C: o nazwie identycznej z rezydentnym odpowiednikiem nie pozwala na użycie tego ostatniego. Nie oznacza to jednak konieczności ich użycia, gdyż opcja REMOVE pozwala na "wyłączenie" wcześniej instalowanego rezydenta. Jeśli posiadamy nowszą wersję komendy RUN (poprawione błędy itp.), możemy zastąpić standardowo wbudowany RUN (system 2.0 lub nowszy) przez naszą komendę.
Resident C:Run lub
Resident Run REMOVE
Użycie drugiej komendy likwiduje wewnętrzny RUN, ale nie instaluje podanego "zamiennika".

Dżokey (znaki globalne).

Podczas niektórych prac z DOSem trzeba wykonać pewną operację na większym zbiorze plików. Bardzo często trzeba np. wydrukować zawartość plików w danym katalogu.
Copy Plik1 to PRT:
Copy Plik2 to PRT:
Copy Plik3 to PRT:
Aby uniknąć takiej "pisaniny" większość systemów DOS wprowadza specjalne znaki umożliwiające wykonanie tej samej operacji na wielu zbiorach. Znaki te zostają przez system zamienione na odpowiednie wzorce, które wyznaczają wybrane pliki. Podstawowe znaki to:

? - Znak dowolnej litery.
- Znak powtórzenia.

W ten sposób nasz problem z drukowaniem można rozwiązać jako:

Copy Plik? to PRT:

gdyż znak *?* będzie przyjmował wszystkie możliwe wartości, a następnie będzie dopasowany do istniejących plików - w naszym przypadku będą to wartości "1", "2" i "3".

W zbiorze:

abcd
afgh
abcf
abcdn
bfcd

do wzorca "abc?" pasują nazwy "abcd" i "abcf". Nazwa "abcdn" jest zbyt długa, natomiast "afgh" i "bfcd" nie pasuje do pierwszych trzech liter wzorca. Z kolei wzorzec "??c?" wybiera nazwy "abcd", "abcf" i "bfcd". Znak zapytania nie może zastępować braku litery, stąd "abc" nie pasuje do "abc?"

Znak "#" pozwala na powtórzenie elementu występującego zaraz po nim. Stąd wzorzec "*rop#uch*" będzie pasował do nazw:

ropch
ropuch
ropuuch
ropuuuch

czyli tych mających dowolną ilość litery "u" wewnątrz "ropucha". Nie będzie jednak pasował do:

ropuluch
ropucuh
rupuch

Wynika stąd pewien wniosek - wzorzec *"#?"* pasuje nie tylko do dowolnej litery, ale również do dowolnej nazwy. Można go również używać w środku wzorca, wtedy nazwy muszą posiadać pozostałe litery, aby do niego pasować:

ro#?ch

pasuje do "roch", "ropuch", "rojdslslslsslch", ale nie do "groch". Najczęstszym zastosowaniem jest wykonanie operacji na wszystkich plikach ("#?") lub na ikonach tych plików (#?.info)

Specyficzną cechą AmigaDOSu jest korzystanie z większych obiektów wewnątrz nazw, które deklarujemy nawiasami. Korzystanie z takich obiektów umożliwia powtarzanie dowolnego ciągu liter np. ro#(puch) pasuje do:

ropuch
ropuchpuch
ropuchpuchpuch

Takie obiekty mają również własny operator alternatywy "|". Jego użycie wewnątrz nawiasów pozwala na podanie różnych możliwych ciągów traktowanych jako obiekt:

"bo(abc|def)" pasuje do

boabc
bodef

Bardzo przydatny jest znak "%", który zastępuje brak znaku. Ma on sens głównie przy wykorzystywaniu nawiasów np. "bum(bo|%)" pasuje do

bum
bumbo

Istnieją jeszcze trzy specjalne znaki, dostępne jedynie w systemie 2.0 lub nowszym:

"-" - operator przejścia
"~" - operator negacji
"*" - równoważny "#?"

Znak minus użyty wewnątrz nawiasów klamrowych oznacza wybranie liter z pewnego zakresu. Jeśli wybierane pliki numerujemy lub dodajemy im np. kolejną literę alfabetu, to korzystając z tego znaku możemy wykonać pewną operację tylko na wybranych. Mamy oto pliki:

ListA 1Cena
ListB 2Cena
ListC 3Cena
ListD 4Cena

Podane niżej wzorce wybiorą pliki:

List[B-D] wybiera: ListB, ListC, ListD
[3-4]Cenawybiera: 3Cena i 4Cena

Tylda "~" jest najczęściej używana w połączeniu z grupami (zaznaczane nawiasami "(" i ")", gdyż oznacza blokowanie jakiejś kombinacji. Potrzebujemy kopii jakiegoś katalogu, ale bez ikonek dołączanych do pliku. Wystarczy podać wzorzec "~(#?.info)" i sprawa jest załatwiona!

Jak korzystać z dżokerów?

1). Amigowskie dżokery działają w odniesieniu do wielu programów z katalogu C:, działają również w połączeniu z innymi programami.

2). W systemie 2.0 i 3.0 dodano bardzo przydatny znak "*", równoważny "#?". Pasuje on do dowolnego ciągu liter pozwalając np. kopiować całe katalogi. Oczywiście można go łączyć z pozostałymi znakami, pozwalając na dowolną selekcję. Znak ten trzeba przed użyciem uaktywnić, gdyż czasami powoduje drobne kłopoty. Commodore nie zapewniło takiego programu, stąd warto go poszukać na dyskach PublicDomain np. StarClick Chrisa Hamesa (autor programu PCTask).

3). Radzę samodzielnie przećwiczyć używanie znaczków, gdyż często zachowują się niezbyt zgodnie z przewidywanym efektem. Dotyczy to głownie znaku "~".

4). Znak negacji został wprowadzony dopiero w systemie 2.0.

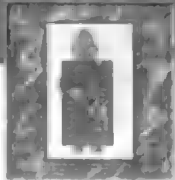
5). Gwiazdka używana zamiast "#?" może czasem doprowadzić do kolizji z jej innym znaczeniem jakim jest aktualne okno. Warto więc zawsze pisać dwie gwiazdki "**", aby uniknąć konfliktów.

6). Znak alternatywy "|" nie działa prawidłowo bez użycia nawiasów.

7). Znaki globalne mogą wskazywać fragment ścieżki, jednak można z tego korzystać w pełni jedynie z systemem 2.0.

8). Bez polskiej lokalizacji oraz formatowania dyskietki w systemie International polskie litery nie będą poprawnie sortowane z użyciem znaku "-".

Na dzisiaj to już wszystko. W przyszłym miesiącu kolejna część rozważań o Amiga Dosie. ❑



Pisma, pisadełka

Tomasz Kokoszczyński

W dzisiejszym odcinku przygód małego pisarza zajmiemy się napisaniem jakiegoś pisemka użyteczności wyższej. Każdemu przecież zdarza się potrzeba wysłania druku do Walnego Zebrania Zarządu Przedszkola Jaś i Małgosia lub Wielce Ważnego Pana Dyrektora Fabryki Długopisów Kulkowych. Nie będę w tym artykule omawiał zasad pisania takich pisemek, ani tym bardziej ich treści, ale zajmę się stroną czysto wydawniczą. Każdy z nas od czasu do czasu staje się przecież takim małonakładowym wydawcą...

W czym pisać?

Pisać można w każdym programie, do którego da się wstukać litery z widocznym na ekranie efektem. Ostatnio próbowano mnie nawet przekonać, że świetnie nadają się do tego arkusze kalkulacyjne w stylu ProCalca.

My nie będziemy jednak zajmować się matematyką tylko poważnym literatnictwem i skoncentrujemy się na 2 programach:

- CED
- PageSetter

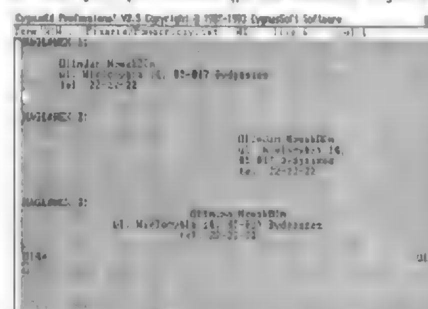
Pierwszy z nich nie jest oczywiście żadnym programem DTP. Może on jednak posłużyć jako punkt wyjściowy dla drugiego, a poza tym to większość prostych pisemek będzie pisana i drukowana głównie przy jego pomocy. Program ten pozwala nam na tworzenie tekstów, które mogą być wysłane bezpośrednio na drukarkę i wydrukowane w trybie tekstowym, co ma szczególne znaczenie przy drukarkach igłowych. Drukarki te mają możliwość drukowania w trybie graficznym i tekstowym. Oczywiście tryb tekstowy stworzono po to, by szybciej drukować teksty. Tryb graficzny jest zaś trybem długotrwałym. Nie ma przecież zawsze takiej potrzeby, żeby z powodu kawałka jakiegoś pisemka przez cały dzień przystosowywać swoje uszy do koncertu heavy metalowego. Nie da się tego oczywiście uniknąć przy programach bardziej specjalizowanych, które uznają tylko druk w trybie graficznym. Swego czasu istniała nawet próbka połączenia obu tych trybów przy pomocy programu PenPal, ale efekty były raczej niezadowalające.

Zacznijmy od nagłówka.

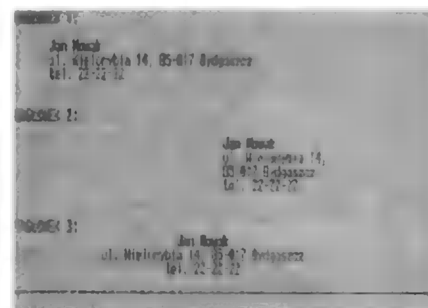
Nagłówek może być ustawiony różnie. Po lewej, po prawej stronie, a czasami pośrodku. To zależy od tego jaką koncepcję przyjmiemy.

Nagłówek po stronie lewej jest najłatwiejszy do ustawienia. Pamiętajmy jednak, że nie musimy zawsze zaczynać od samej krawędzi, a czasami pewien odstęp od brzegu jest nawet wskazany. Odstępy produkujemy najłatwiej przy pomocy tabulatora.

Czasami, w wyniku rozbieżności ustawień w drukarce, może nie wyjść to tak jak powinno (przeważnie wycho



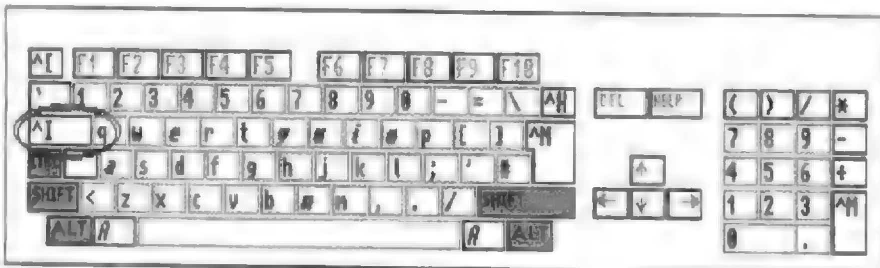
3 typy nagłówków w CEDzie.



I to samo oglądane inaczej.



Można i tak.



Tabulator.

dzi). W takim układzie po prostu przy wydruku z CEDa wstawiamy drukowanie tabulatorów jako spacji.

Środkowe ustawienie nagłówka w CEDzie uzyskujemy kombinacją klawiszy: Amiga + \. Oczywiście działanie tej funkcji zależy od ustawienia prawej granicy tekstu (Set right border). Granica ta powinna być ustawiona tak, żeby tekst wysłany na drukarkę miał swój prawy brzeg dokładnie tam, gdzie sobie tego życzymy.

Przy nagłówku ustawionym pośrodku można sobie wstawić dodatkowo linię poziomą, która wydzieli tekst naszego adresu od reszty listu.

Te [redacted] nagłówki [redacted] PageSetterze.

Postarajmy się teraz dokonać tych samych "cudów" przy użyciu programu PageSetter. W tym celu musimy wczytać tekst do otwartego pudełka. Tylko tu jedna zasadnicza uwaga: jeżeli do PageSettera wczytamy za dużą ilość tabulatorów, to możemy mieć problemy z tekstem - wyskoczy on nam gdzieś poza pudełko i będzie się wydawało, że nie został wczytany. Zdarza się to szczególnie przy prawostronnym ustawieniu tekstu. Tekst, który chcemy mieć ustawiony centralnie ustawiamy ręcznie. Służy do tego celu opcja: Justification/Center z menu Type. Aby z niej skorzystać musimy mieć uprzednio zaznaczony tekst (LMB i przeciągnięcie kursora ponad fragmentem tekstu do zaznaczenia).

Niestety nie udało mi się zmusić PageSettera do przyjmowania systemowych kodów ANSI, które normalnie powodują np. wytłuszczenie, czy też podkreślenie. Dodatkowe efekty [redacted] uzyskać wstawiając do tekstu symbole:

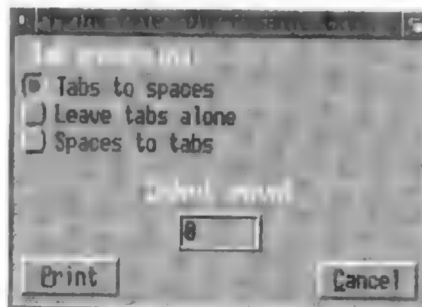
\B lub \b - włączenie lub wyłączenie wytłuszczenia

\U lub \u - włączenie lub wyłączenie podkreślenia

\I lub \i - włączenie lub wyłączenie kursywy

\O lub \o - włączenie lub wyłączenie obrysów.

Ostatnia opcja nie ma odpowiednika w kodach ANSI.

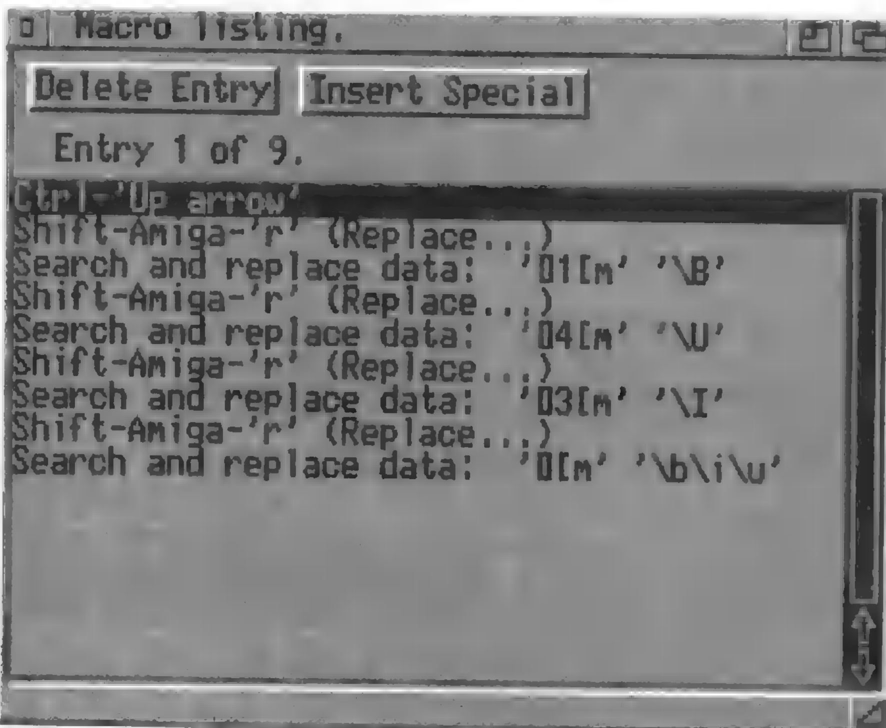


A co zrobić jeśli ktoś sobie przygotował tekst w CEDzie, który ma wiele takich znaków i chciałby to w ostatniej chwili przenieść do PageSettera z zachowaniem wszystkich szykanów? Oczywiście można te znaczki powymieniać przy pomocy funkcji Repeat Replace. Jeśli okaże się, że jesteśmy zmuszeni wiele razy korzystać z takiego trybu działania, powinniśmy utworzyć sobie w CEDzie odpowiednie makro.

Chwilka o makrach CEDa

Makra są to zestawy instrukcji wykonywanych po kolei tak, jak to miało miejsce przy ich wprowadzaniu. Zaczynamy od wybrania podmenu "Begin short invocation macro" z menu "Environment/Macro definitions". Teraz musimy wybrać sobie klawisze, przy pomocy których będziemy to makro wywoływać. Niech to będzie np. Amiga + /. Makro takie wygląda tak, jak to przedstawiono na rysunku.

Po kolei mamy tu: skok do początku tekstu i wymianę poszczególnych znaczków na inne (lepsze). Oczywiście uniwersalne wyłączenie tych funkcji w CEDzie (ESC[0m) podstawiamy trzema wyłączeniami PageSettera (\b\i\u), bo nigdy przecież nie wiemy, na które trafiliśmy. Jest to nieco niewygodne, gdyż można czasami wyłączyć nie to co trzeba. W tym celu możnaby oczywiście napisać program w ARexxie, ale w chwili obecnej nie będziemy się tym zajmować, bo "skórka nie jest warta wyprawki".



Makro CEDa wymieniające "bajery".



Kolejne strony

Tekst listu, czy pisma oczywiście musimy wymyślić sami. Problem zacznie się wtedy, gdy będzie go więcej i zaistnieje konieczność ponumerowania stron.

W tym celu w CEDzie musimy zorientować się ile linii wychodzi na kartkę i wtedy możemy co tyle linii wstawić numer strony. Tego nie da się już tak łatwo wykonać przy pomocy makra. Praktyczniejszy jest tu programik ARexxa, który kiedyś mi się bardzo przysłużył:

```
/* This ARexx program set page # - 1 */
/* page = 61 lines */
cr = '0A'X

options results

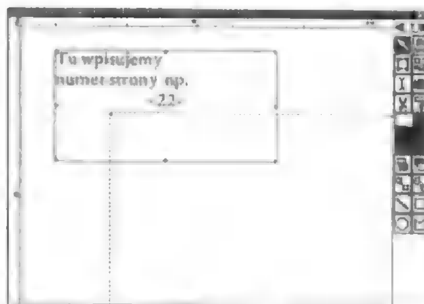
ADDRESS 'rexx_ced'

'Beg of file'
linia = -1
strona = 1
do i = 1 to 20
linia = linia + 62
strona = strona + 1
jump to line... linia
status 57
if result < linia-1
      then exit 0
else call ZrobTo
end
exit 0

ZrobTo:
Beg of line
text '                    ||strona||' cr cr
return 0
```

Program ten wstawia automatycznie numery strony co 61 linii. Rozstaw spacji w numerze strony jest dobrany do szerokości tekstu 65 linii, ale może być przecież dowolnie zmieniany. Oczywiście tekst nie będzie nam się idealnie zazębiał ze stronami i dlatego konieczne jest późniejsze jego sformatowanie, tzn. takie poustawianie akapitów, żeby wszystko wyglądało ładnie. Polecam uruchomienie programiku i następne skorzystanie z funkcji undo. Linie w których trzeba coś zmienić możemy sobie zaznaczać markerami (Amiga + Shift + 1 lub 2 lub 3) i po wykonaniu cofania oznaczenia stron wrócić do nich żeby wyrównać akapit, czy też wyciąć jakiś niewygodny wyraz.

Niestety PageSetter nie pozwala na automatyczne numerowanie stron. Wszyscy jego nowsi koledzy robią to bez zmrużenia oka, natomiast on nie potrafi. Tak więc w poważniejszych programach wystarczy wpisać odpowiedni zestaw znaków do dowolnego pudełka, a ukaże nam się numer strony. Natomiast jeśli chodzi o PageSettera trzeba sobie samemu stworzyć dodatkowe pudełko i do niego wpisać numer strony. Czasami jest to na tyle monotonne i uciążliwe, że można zapomnieć zmienić numer strony i wtedy będziemy mieli kilka stron o numerze np. 3. W przypadku programów DTP numery stron możemy wpisywać w dowolnych miejscach - na środku u góry, na dole, na dole przy brzegu strony. Starajcie się jednak unikać takich podstawowych błędów, jak umieszczenie numeru strony przy brzegu wewnętrznym - wtedy, żeby go dojrzeć, trzeba prawie rozerwać książkę czy pismo. Nie należy także wpisywać numeru zbyt wielką czcionką. Jest to w sumie tekst pomocniczy i jeśli komuś będzie potrzebny, to sobie go znajdzie. Nie powinna zaistnieć przecież taka sytuacja, że numer będzie większy od tytułu. Ważniejszy jest chyba tytuł, czy nawet sam tekst.

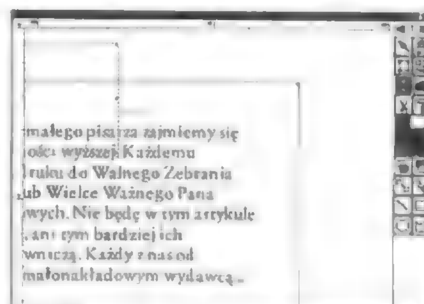


Niby można ustawić byle gdzie...

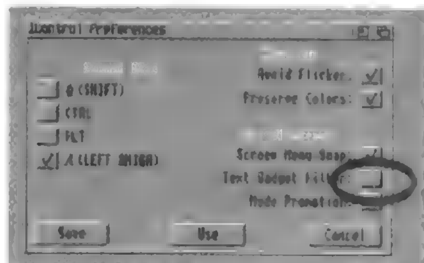
Pod linijkę

Gdy już ponumerujemy sobie strony i wstawimy nagłówek to przychodzi moment na wczytanie właściwego tekstu (przeważnie wczytuje się on od razu). Tekst ten może mieć brzegi zazębione albo ułożone pod linijkę. W CEDzie ułożenie pod linijkę następuje przy użyciu opcji Format/With Fill z menu Special. Prościej jest wywoływać tę opcję przez wciśnięcie kombinacji klawiszy Amiga + f. Wystarczy przytrzymać te klawisze, a wyrównaniu ulegną wszystkie akapity. Mogą być one oddzielone od siebie pojedynczą linią lub po prostu zaczynać się od znaku tabulatora, czy spacji. CED to wszystko Wam uwzględni.

Gorzej jest w przypadku programów DTP. Te wymagają wstawienia tekstu, którego akapity nie mają linii posiekanych returnami. Wstawienie tekstu napisanego w CEDzie spowoduje tragiczne w skutkach cięcia w środku akapitu.

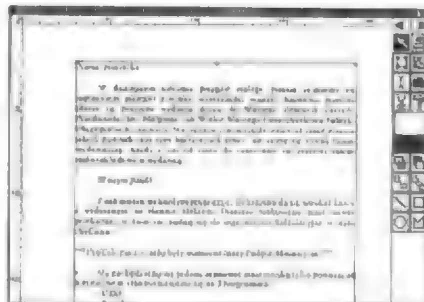

Tak wyglądają "nieprzystosowane" linie.

Żeby tego uniknąć, należy wszystkie końcówki linii czyli znaki Enter w środku akapitu pozamieniać na spacje. Umożliwia to stworzony kiedyś przeze mnie program Texton opcja LongLines (automatyczne rozpoznawanie akapitów), ale można też tego dokonać od biedy ręcznie przy pomocy CEDa. Pamiętajmy, że znak Enter uzyskujemy w requesterze szukania poprzez wciśnięcie Ctrl-J. Aby ta operacja była skuteczna w wyższych systemach musimy zwrócić uwagę na ustawienie poniższej opcji w preferencjach.

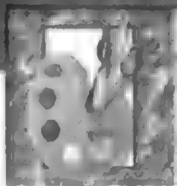


Text Gadget Filter musi być wyłączony.

Czasami jednak przy dłuższych liniach CED będzie się biesił twierdząc, że linia jest za długa. Na to akurat nie ma rady. Albo skrócimy akapit, albo ręcznie w programie połączymy dwie jego części.


Text Gadget Filter musi być wyłączony.

Na dziś wystarczy. Myślę, że teraz przyjdzie do redakcji więcej ciekawych listów. Cześć. ❑



IMAGINE cz.4

W tym odcinku jeszcze trochę teorii na temat tworzenia [illegible] Imagine. Tym razem zajmiemy się szkłem. Wbrew pozorom jest ono trudne do uzyskania i zazwyczaj trzeba niezliczonej ilości prób, aby uzyskany efekt był zgodny z rzeczywistością.

Postanowiłem, że pierwsze logo będzie wykonane tylko przy użyciu dwóch programów tzn. DeLuxe Painta i Imagine'a. Są one dostępne dla każdego i nie sprawiają kłopotu na poczciwej pięćsetce z jednym megabajtem pamięci. Co prawda istnieje kilka programów specjalnie napisanych, aby ułatwić obróbkę i konwersję obiektów np. PIXEL 3D, VERTEX 2.0, INTER CHANGE PLUS, lecz wymagają one większej pamięci i twardego dysku. Może w najbliższym czasie kogoś namówię do ich opisania i porównania. Czasowo ograniczam także ilość rysunków w tekście. Jakość druku powoduje, że stają się one nieczytelne, tak więc będzie ich mniej, ale za to większych. OK, to tyle gadania, bierzemy się do roboty.

Jak stworzyć szkło.

Szkło jest chyba najtrudniejsze do uzyskania. Jest to wartościowy atrybut, ale wymagający pewnych trików, aby końcowy efekt był perfekcyjny. Pierwszym problemem jest w ogóle uzyskanie tego efektu. Najczęściej słyszanym narzekaniem jest: "ja nie mogę uzyskać niczego przezroczystego". Powodem tego jest błąd Imagine'a. Jakakolwiek połyskliwość w atrybutach obiektu obniża przezroczystość. Musisz używać połyskliwości równej 0. Nie małej, lecz zerowej. Tylko wtedy może coś zadziałasz.

Przezroczystość jest sterowana przez atrybut filtru. Im wyższa wartość, tym więcej światła danego koloru przechodzi. Aby całkowicie odfiltrować światło, musisz ustawić wartość 0 0 0. Dobra przezroczystość wymaga dużej amplitudy. Przy mniejszych wartościach efekt nie jest widoczny. Dla obiektów przezroczystych najmniejsza wartość, której używam, wynosi 200, ale możesz eksperymentować. Obiekty dobrze prześwitujące wymagają wartości od 240 do 255. Poza tym, szkło ma niebieskawy odcień, więc używam RGB= 250,250,255.

Szkło bardzo dobrze odbija światło, ma bardzo jasne odblaski. Ja ustawiam specular na 255,255,255 i hardness na 255. Miękkie odblaski wyglądają źle, a poza tym zmniejszają przezroczystość.

Jeśli chcesz użyć szkła nie zapomnij o współczynniku załamania, który mówi jak bardzo światło ugnie się przy przejściu przez granicę dwóch ośrodków. Im większy współczynnik, tym większe ugięcie. Wartość 1,0 oznacza brak ugięcia i tak jest w powietrzu. Wartość 2,9 oznacza prawie nierzeczywiste ugięcie światła.

Pamiętaj, że jeżeli obiekt będzie przezroczysty ze współczynnikiem załamania wynoszącym 1,309, nie uzyska efektu obiektu zrobionego z lodu! Inne atrybuty są prawie tak samo waż[illegible] do nadania przezroczystym obiektom ich charakteru. Poza tym, przy zbyt dużym współczynniku załamania przechodzące światło będzie tak ugięte, że obraz nie będzie rozpoznawalny. Tyczy się to dużych lub skomplikowanych obiektów. Wtedy mniejsze współczynniki załamania dają lepszy efekt (i szybciej się generują!). Wszystko co jest przezroczyste staje się soczewką, [illegible] soczewka jest kiepskim przyrządem optycznym. Dla przezroczystej kuli musiałem obniżyć współczynnik załamania do 1,08, aby obiekty po drugiej stronie były rozpoznawalne.

Chropowatość i mapa wysokości są szczególnie efektowne przy obiektach przezroczystych. Chropowata powierzchnia umożliwia uzyskanie przy obiektach przezroczystych wielu ciekawych efektów. Wyobraź sobie świeży sześcian lodu - jest on przezroczysty, lecz szron powoduje, że trudno jest cokolwiek przez niego zobaczyć. Gdy szron zniknie, powierzchnia staje się gładka i dobrze przezroczysta.

Proponuję używanie przypadkowej mapy wysokości tworzonej za pomocą airbrusha [illegible] DPaint III, niż używanie atrybutu roughness. Jest to spowodowane tym, że przy roughness uzyskuje się różne zmiany na powierzchni (tak powinno być), lecz zmieniają się one co klatkę podczas animacji, co wygląda jakby po powierzchni łaziło mnóstwo robaków.

Lista współczynników załamania.	
Powietrze	1,02
Lód	1,309
Alkohol	1,329
Woda	1,333
Szkło	1,50
Kwarc i sól	1,644
Diament	2,417



Kierunek powierzchni jest bardzo ważny przy przechodzeniu przez nią światła, więc równie ważne jest cieniowanie Phonga. Wygładza ono obiekty złożone z wieloboków, aby nie wyglądały jak oszlifowane kamienie szlachetne. Cieniowanie Phonga jest używane do określenia kierunku, w którym ugina się światło.

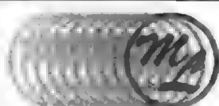
Kolor szkła określa, jaki odcień będzie dawało niezupełne szkło - aby uzyskać szkło nieprzezroczyste ustaw kolor na 255 255 255. Kolor czarny będzie natomiast dobry (0,0,0). Możesz eksperymentować.

Ostatnim atrybutem szkła jest odbicie (reflection). Szkło nieznacznie odbija światło, więc powinno być nieco odbijające. Wyobraź sobie okno. Jeśli nie ma odbić, to widzisz przez nie wyraźnie. W nocy, kiedy przez okno przechodzi mało światła, zaczynasz widzieć jego odbijające (lustrzane) własności. Dobra wartość odbić wynosi 30-60.

Ciekawym zagadnieniem są soczewki. Możesz je stworzyć i będą one działały! Aby stworzyć prostą soczewkę, stwórz kulę o większej ilości sekcji (dwa razy tyle co wartość domyślna). Przejdź do trybu "select points" i zaznacz wszystko poza górnym 20% kuli. Skasuj te punkty. Przesuń oś na sam dół półsoczewki przy użyciu M (shift-M). Upewnij się, że współrzędna Z osi jest jak najbliżej (ważne) współrzędnej z dolnego pierścienia punktów. Następnie wybierz obiekt, skopiuj go (COPY) i wstaw (PASTE). Teraz są już dwie identyczne połowy, jedna na drugiej. Wybierz jedną i przeskaluj przy użyciu transformacji do x=1,0, y=1,0, z=-1,00. Jeśli oś była umieszczona prawidłowo, otrzymasz dwie połówki soczewki o wspólnym środkowym rzędzie punktów. Wybierz obie połówki i połącz je (JOIN) w jeden obiekt, a następnie usuń zbędne punkty przez MERGE. Ustaw atrybuty na szkło i... Działa! Mamy szkło powiększające. Możesz spróbować stworzyć szkło pomniejszające. Szkło powiększające może mieć o wiele większy współczynnik załamania bez zaciemniania obrazu. Zasady - obiekt daleki będzie do góry nogami, obiekt zbyt blisko będzie bardzo duży, obiekt w odległości ogniskowej będzie w ognisku i powiększony. Ogniskowa jest proporcjonalna do r (promienia kuli) i współczynnika załamania. Trochę skomplikowane, ale ciekawe. Na koniec:

Przezroczysta kula
Color =0 0 0
Transp= 250 250 255
Reflec= 49 49 53
Specular= 150 150 150
hardness= 255 255 255
rough=0
shininess=0 (CRITICAL)
Index=1.08

To tyle na dzisiaj. Serdecznie podziękowania panu Steve Worley'owi za pomoc w zrozumieniu jak się robi prawdziwe szkło, a także Arturowi za pomoc przy opracowaniu tego tekstu. ❑



Micro-Luc

40-008 Katowice ul.Wodna 1/4
tel.(0-32) 1538-503

AKCESORIA
do komputerów AMIGA

0.5 MB RAM EXPANSION	350000
2.0 MB RAM EXPANSION (0.5 MB chipRAM + 1.5 MB slowRAM)	1450000
ZEGAR plus Akku do modułu RAM	180000
Videodigitizer MicroVIEW	1400000
KickStart 2.04 dla A500	650000
KickStart 1.3 dla A500+ i A600	470000
Amiga TURBO SYSTEM	1300000
Sampler stereo VOICE 100 kHz	650000
BootSelector elektroniczny	200000
MIDI Interface	330000
Spliter RGB	1300000
4.0 MB RAM dla A1200	5500000

Do podanych cen należy dodać 22% podatku VAT

Prowadzimy sprzedaż wysyłkową !

Wystarczy gdy wpłacisz pieniądze na konto, a dowód wpłaty prześlesz na nasz adres.

Szczegółowych informacji udzielamy w siedzibie firmy.

Nasze konto: PKO-BP II o/K-ce 27528-737212-136

Dla odbiorców hurtowych - RABATY

InterComp

Już w sprzedaży:

AMIGA CD32
AMIGA 1200
AMIGA 4000

Oferujemy także nasze programy

Virus Expert 2
Polskie litery do PageStreama!!!
Księga Przychodów i Rozchodów VAT!!!
Wypożyczalnia kaset video!!!
Hydacom - gra logiczna z bajerami!!!

Oferujemy też

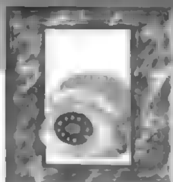
wszelkie peryferia do komputerów AMIGA jak:
rozszerzenia pamięci
twarde dyski
dowolne karty do komputerów A2000/3000/4000
CD-ROMy do komputerów A500/2000/3000/4000

Specjalna oferta dla profesjonalistów:

szeroka gama komputerów, kart graficznych, kart turbo [illegible] sprzętu niezbędnego do prac video.

U NAS kupisz m.in.:

Karty Impact Vision w wersjach S-VHS i Beta
Karty Retina (tania 24-bitowa karta wraz z oprogramowaniem)
Karta Emplant (emulator Macintosha - pracuje w kolorze)
Licencjonowane oprogramowanie graficzne i użytkowe.
m.in. SCALA, Imagine, Real 3D, Final Copy II, SAS C, Pascal, PageStream, ADPro, Image FX, SBPro, Art Expression, Pixel Pro, [illegible] Link, Personal Paint, DynaCAD, Directory Opus i inne.



MODEMOWANIE cz.1

Adam Grzymowicz

W ostatnich latach można zaobserwować tręd do stworzenia ogólnoświatowej wioski elektronicznej. Jednym z podstawowych postulatów tej koncepcji jest niczym nieskrępowana wymiana informacji między mieszkańcami różnych zakątków Ziemi. Do tego celu wyśmienicie nadają się modemy...

Tak, właśnie urządzenia zwane modemami umożliwiają wszystkim ich posiadaczom wymianę danych, programów i listów. Dzięki tzw. poczcie elektronicznej można podyskutować z wieloma interesującymi osobami, nawet z innych krajów, i wcale nie trzeba dzwonić za granicę, co podejrzewam leży poza możliwościami finansowymi większości z Was. Ale o tym później. Na początek zajmijmy się podstawami, czyli MODEMEM.

Co to jest modem? Modem to "czarna skrzynka" służąca do przesyłania danych pomiędzy dwoma komputerami za pośrednictwem łączy. Łączami tymi mogą być specjalnie położone kable lub po prostu zwykłe linie telefoniczne. Te ostatnie mają tę ogromną zaletę, że istnieją i to w bardzo dużych ilościach, łącząc miasta, kraje i kontynenty. Dokładna zasada działania modemu nie ma dla nas większego znaczenia, warto jednak wiedzieć skąd pochodzi jego nazwa. Otóż, sygnał cyfrowy przekazywany z komputera za pomocą łącza szeregowego trafia do modemu nadawczego, który moduluje go na postać analogową dostosowaną do standardowej linii telefonicznej i wysyła w tę linię. Modem odbiorczy zaś demoduluje sygnał odebrany od modemu nadającego i już w postaci cyfrowej przesyła do komputera odbierającego informację. Ponieważ przy przekazywaniu danych zachodzi MOdulacja i DEModulacja, stąd też nazwa MODEM.

Dawno, dawno temu na świecie królowały modemy łączone z linią telefoniczną akustycznie. Polegało to na tym, że modem posiadał mikrofon i głośnik, które nakładało się na słuchawkę (fachowa nazwa to mikrotelefon) dowolnego aparatu telefonicznego. Na pierwszy rzut oka było to bardzo wygodne rozwiązanie - nie trzeba było nic podłączać, lecz nie pozwalało ono uzyskiwać wysokich prędkości transmisji. Jednocześnie w Polsce modemy tego typu są w zasadzie bezużyteczne, gdyż modem potrafiący jedynie piszczeć do słuchawki, nie jest w stanie wykręcić numeru. Spowodowane jest to stosowanym u nas wybieraniem impulsowym, polegającym na zwieraniu linii w określonych odstępach czasu. W krajach gdzie stosowane jest wybieranie tonowe, modemy akustyczne nie mają takich problemów. Ponieważ jednak mieszkamy w Polsce, będziemy musieli używać tak zwanych modemów galwanicznych, czyli takich, które są bezpośrednio podłączane do sieci telefonicznej (np. przez wetknięcie wtyczki). To właśnie te modemy są właściwymi modemami (modemy akustyczne nazywane są najczęściej sprzęgami akustycznymi), których używa cały świat.

Zanim jednak wybierzemy się do sklepu, aby kupić modem, warto wiedzieć o jeszcze jednym fakcie. Modemy dzielą się na zewnętrzne i wewnętrzne. Te pierwsze komunikują się z komputerem za pośrednictwem łącza szeregowego (popularnego RS232) i mogą współpracować z każdym typem komputera posiadającym złącze RS232. Natomiast modemy wewnętrzne (jako karta do komputera) są przeznaczone głównie do komputerów kompatybilnych z IBM, czyli dla nas zupełnie nieprzydatne. Jak do tej pory jedyną firmą produkującą modemy na karcie dla Amigi jest Supra, ale żeby jej użyć, trzeba być posiadaczem A2000, A3000 lub A4000. Modemy zewnętrzne są z reguły droższe od modemów wewnętrznych o takich samych możliwościach, gdyż w cenie modemu zewnętrznego zawarty jest koszt obudowy i zasilacza. Jednak my, posiadacze A500/600/1200 nie mamy innego wyjścia i musimy stosować modemy zewnętrzne. Oprócz tego niektóre z modemów posiadają możliwość nadawania i odbierania dokumentów przesyłanych faxem. Modemy te - zwane faxmodemami - wraz z odpowiednim oprogramowaniem zastępują w pełni telefax.

Wiadomo już więc, że potrzebny nam jest modem galwaniczny (modemy akustyczne są i tak w Polsce niespotykane) i zewnętrzny, a jeśli mamy zamiar przesyłać faxy, to faxmodem. Problemem pozostaje, który z dostępnych modeli najlepiej wybrać. Niezłym sposobem na rozstrzygnięcie wątpliwości jest wzięcie pod uwagę kilku cech:

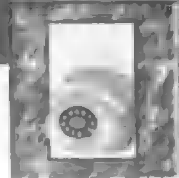
- prędkości transmisji,
- korekcji błędów i kompresji,
- kompatybilności z faxami (dla faxmodemów).

Prędkość transmisji określana jest na ogół w bitach na sekundę, a poszczególne zakresy prędkości zostały unormowane przez Międzynarodowy Komitet Doradczy Telegrafii i Telefonii (CCITT) i oznaczone są następująco:

V.21 - 300 bps,
V.22 - 1200 bps,
V.22Bis - 2400 bps,
V.23 - 1200 bps z kanałem zwrotnym 75 bps,
V.32 - 4800, 9600 bps,
V.32Bis - 7200, 12000, 14400 bps.

Niektórzy producenci modemów zaprojektowali własne protokoły, umożliwiające uzyskiwanie większych prędkości niż 14400. Należy do nich protokół HST (firmy US Robotics), który umożliwia uzyskanie prędkości 16800 bps (ostatnio 21600 bps) w jednym kierunku i 450 bps w drugim. Modem, który chce nadać dużą ilość danych otrzymuje dostęp do prędkości 16800. Kolejnym protokołem jest ZyXEL umożliwiający połączenia z prędkości 16800, a także 19200 bps. Kolejny to PEP (zaprojektowany przez Telebit) pozwalający uzyskiwać 20000 bps. Modemy pracujące z niestandardowymi (nie zatwierdzonymi przez CCITT) protokołami mogą je wykorzystać podczas połączenia z modemami pracującymi z tym samym protokołem. Chociaż oczywiście mają one możliwość korzystania z innych zatwierdzonych przez CCITT standardów. Jednak przed kupnem lepiej się o tym upewnić.

Jaka prędkość nam wystarczy zależy od tego, do czego będziemy wykorzystywali modem. Jeśli będziemy się ograniczać tylko do poczty elektronicznej i przesyłania krótkich plików, to wystarczy nam w zupełności prędkość 2400 bps. Niższe prędkości są



mało użyteczne, gdyż duża część polskich BBSów (będę jeszcze o nich pisał) nie pozwala na połączenie z prędkością 1200, nie mówiąc już o [illegible] bps.

Gdy będziemy chcieli przesyłać duże ilości długich programów, nie obejdzie się bez szybszego modemu. Przykładowo - przesłanie [illegible] MB [illegible] pomocą modemu 2400 zajmuje około 70 minut, natomiast [illegible] wykorzystaniem modemu 14400 już tylko 10-12 minut.

Innym kryterium jest korekcja błędów i kompresja danych. Przy większych prędkościach transmisji - zwłaszcza na nienajlepszych liniach - staje się bardziej wrażliwa na różnorakie zakłócenia na linii. Stąd konieczność stosowania układów korekcji błędów. Jeśli dodatkowo dane będą kompresowane, to transmisja zajmie mniej czasu i mniejsze będzie prawdopodobieństwo jej zakłócenia. Obecnie spotykane protokoły kompresji danych i korekcji błędów to:

Kompresja: MNP-5 - kompresja 2:1, MNP-7 - kompresja 3:1, V.42Bis - kompresja 4:1.

Korekcja błędów: MNP-5, MNP-3, MNP-4, V.42.

Metody kompresji i korekcji o nazwach MNP są rozwiązaniami amerykańskiej firmy Microcom (MNP - Microcom Networking Protokol).

MNP-2, MNP-3 - obecnie rzadko spotykane, polegały na potwierdzaniu przez modem odbioru każdego przesyłanego znaku.

MNP-4 - polega [illegible] transmisji w blokach. Sprawdzeniu podlega nie pojedynczy bajt, lecz cały blok.

MNP-5 - bardzo popularny. Używany razem z MNP-4 służy do kompresji danych. Za pomocą MNP-5 można uzyskać nawet dwukrotne ich upakowanie. Dotyczy to jednak tylko plików niespakowanych przed transmisją. W przypadku przesyłania danych już skompresowanych, MNP-5 może wydłużyć czas transmisji.

MNP-6 - umożliwia dobranie przez modem prędkości transmisji w zależności od jakości linii telefonicznej.

MNP-7 - protokół kompresji danych, pozwala na trzykrotne skrócenie czasu transmisji. Dosyć żadko spotykany.

Pozostałe protokoły zostały opracowane przez CCITT:

V.32Bis - pozwala na automatyczną zmianę szybkości transmisji w przypadku połączenia o złej jakości.

V.42 - sposób korekcji błędów, bardziej niezawodny niż MNP-4. V.42 jest kompatybilny "w dół" z protokołem MNP-4.

V.42Bis - protokół kompresji danych, ostatnio bardzo popularny. Jego stosowanie umożliwia nawet czterokrotne spakowanie danych. W odróżnieniu od MNP-5, protokół V.42Bis śledzi przesyłane dane i gdy zauważy, że [illegible] dają się one zbytnio kompresować wyłącza kompresję.

Kończąc problem kompresji i korekcji należy wiedzieć, że najczęściej spotykane w modemach [illegible] protokoły NMP-5 i V.42Bis. Jeśli więc chcemy mieć urządzenie z korekcją błędów i kompresją, musimy szukać modemu właśnie z MNP-5 lub V.42Bis (pamiętając o tym, że właściwie wszystkie modemy mające V.42Bis, mają także MNP-5).

Pozostał już tylko problem zgodności z faxami. Sposoby działania faxów zostały również ujednolicone przez CCITT:

V.21 - kanał dla faxów o prędkości 300 bps,
V.27 - dla prędkości 2400, 4800 bps,
V.29 - dla 7200 i 9600 bps,
V.37 - prędkość 14400 bps.

Wybór właściwego modelu faxmodemu jest dosyć trudny, ponieważ sam faxmodem bez odpowiedniego programu nie na wiele nam się przyda. A przecież - niestety - większość oprogramowania sprzedawanego z modemami przeznaczona jest na IBMy. Jeśli więc nie uda nam się zdobyć odpowiedniego programu, posiadając nawet bardzo dobry faxmodem, niewiele będziemy mogli z nim zrobić. Dobrym wyjściem jest kupno faxmodemu firmy SUPRA, która sprzedaje modemy wraz z oprogramowaniem na Amigę (np. SUPRA Fax-Modem 14400 opisywany w Amigowcu 2/94).

Czasami warto zwrócić uwagę [illegible] cenę produktu. Ceny modemów kształtują się różnie w zależności od możliwości konkretnego modelu oraz dołączanego (bądź nie) oprogramowania. Modemy mieszczą się w przedziale cenowym od 150 DM za najbardziej prymitywne (2400 bps) do powyżej 1000 DM za najbardziej wymyślne (16800 bps, itd.).

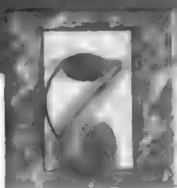
Gdy zdecydowaliśmy się już [illegible] konkretny model i właśnie zostaliśmy jego właścicielem, czas podłączyć go do komputera.

Do tego celu będziemy potrzebowali odpowiedni kabel. Ponieważ komunikacja między komputerem, [illegible] modemem odbywa się za pośrednictwem łącza szeregowego, będzie to kabel posiadający z jednej strony 25-stykowe żeńskie gniazdo (wtykane w port RS232 w komputerze), a z drugiej 25-stykowy męski wtyk (wtykany w gniazdo w modemie). Kabel taki można kupić w sklepie, lecz nie należy kupować kabli nazywanych null-modem. Kable te wbrew nazwie nie służą do podłączenia modemu do komputera. Są to kable do łączenia dwóch komputerów przez port szeregowy i posiadają na obydwu końcach 25-stykowe złącza żeńskie. Kabel do podłączenia modemu można dosyć łatwo wykonać samemu. Wystarczy kupić 25-stykowy wtyk męski oraz żeński i połączyć je kablem. Do tego celu nie jest wymagany 25-żyłowy kabel, gdyż do poprawnej pracy modemu jest potrzebnych znacznie mniej sygnałów. Zresztą na łączu szeregowym Amigi wykorzystanych jest tylko trzynaście pinów. Z tego dwa, to napięcia +12V i -12V (piny 9 i 10), kolejne dwa, to wejście sygnału analogowego (pin numer 18) podawanego na wejście prawego przedwzmacniacza dźwięku zawartego w Amidze i wyprowadzonego na prawe wyjście dźwięku oraz wyjście sygnału audio (pin 11) poprowadzone z lewego wyjścia dźwięku z Amigi. Do komunikacji z modemem używane są piny o numerach 2, 3, 4, 5, 6, 7, 8, 20, 22, więc wystarcza dziewięciożyłowy kabel. Ponieważ wszystkie złącza mają wytłoczone koło styków numerki, łączenie jest bardzo proste i polega na połączeniu styku z numerem 2 we wtyku męskim, ze stykiem 2 w złączu żeńskim, styku 3, ze stykiem 3, 4 z 4, itd.

Po połączeniu modemu z komputerem pozostaje już tylko podłączenie modemu do sieci telefonicznej. W większości modemów do łączenia z siecią telefoniczną służą gniazda i wtyki amerykańskiego standardu RJ11. Na początek trzeba więc zakupić przelotkę z polskiego wtyku telefonicznego na standard RJ11. Oczywiście można zastosować tak zwane metody partyzanckie i przykręcić kabel modemu bezpośrednio do krajowego złącza telefonicznego. Po uporaniu się z tym problemem podłączamy modem do sieci telefonicznej za pomocą dołączonego do modemu kabla, wtykając go w modemie w gniazdo oznaczone "Line" (lub jakoś podobnie - radzę przeczytać instrukcję modemu). Natomiast do gniazda w modemie oznaczonego "Phone" podłączamy telefon, który dotychczas był podłączony do gniazda telefonicznego. Tu również musimy pokonać problem amerykańskiego gniazda. Taki sposób podłączenia telefonu ma tę zaletę, że w trakcie nawiązywania łączności za pośrednictwem modemu, odłącza on telefon od linii i nie musimy słuchać podzwaniania telefonu w czasie, gdy modem wykręca numer. Jednak nie wszystkie modemy posiadają możliwość przelotowego podłączenia telefonu. Wówczas telefon podłącza się równolegle z modemem do tego samego gniazda telefonicznego.

Z podłączeniem modemu do sieci telefonicznej wiąże się jeszcze jedna kwestia - homologacja. Homologacja to certyfikat stwierdzający zgodność parametrów danego typu modemu z wymogami krajowej sieci telefonicznej. Wystawieniem homologacji nie musi zajmować się każdy użytkownik modemu, należy to do obowiązków firmy rozprowadzającej modemy w danym kraju. Za używanie modemu bez homologacji grozi kara na rzecz administratora łączy, tak więc w naszym interesie leży kupno modemu homologowanego, gdyż po pierwsze będziemy mieli wtedy mniej kłopotów [illegible] otrzymaniem dobrej jakości transmisji (modemy, które nie otrzymały homologacji nie są dopasowane do polskich linii, więc mogą mieć problemy ze współpracą z nimi lub niszczyć je - zazwyczaj opinia administratora) i po drugie nie narazimy się na niepotrzebne nieprzyjemności. Obowiązkowe jest również zgłoszenie posiadania modemu do operatora sieci (w naszym przypadku Telekomunikacja Polska S.A.) i uiszczenie opłaty abonamentowej (tak jak za telefon).

Mamy już modem podłączony i do komputera i do sieci telefonicznej. Aby móc go używać, musimy uruchomić odpowiedni program komunikacyjny. Ale o tym następnym razem. ❑



AMIGA i muzyka



Marcin Gackowski

Każdy z nas wie jak ściągi ułatwiają życie. Co więcej, czasami są one niezbędne. I właśnie w tym odcinku kącika muzycznego chciałbym jedną z takich niezbędnych ściąg przedstawić.

Owe ułatwienie to nic innego jak tabela akordów. Niby banalna rzecz, a naprawdę niesamowicie ułatwiająca pracę. Z pomocą takiej tabeli stworzenie linii basu czy ułożenie motywu jest o wiele łatwiejsze niż bez niej. Oczywiście nie zawsze linia basu składa się tylko z dźwięków składowych akordu, ale to już inna historia.

Tabela zawiera składowe dźwięki dla najbardziej popularnych typów akordów. Proponuję umieścić ją obok monitora.

Nie zapominajmy, że dla Protrackera dźwięk B = A#, a dźwięk H = B. Wynika to z niezgodności w nazewnictwie dźwięków. Tabela sporządzona jest według systemu obowiązującego w Polsce (C C# ... A B H), a Protracker pracuje w systemie przyjętym na zachodzie (... A A# B).

Tyle tytułem wyjaśnień. Teraz na podstawie paru prostych przykładów chciałbym pokazać użyteczność tabeli...

Potrzebne nam będą (oprócz Protrackera oczywiście):
1 - stopa,
2 - werbel,
3 - chórek moll,
4 - chórek dur,
5 - hihat,
6 - bas,
7 - piano

To oczywiście bardzo prymitywny zestaw instrumentów, chodzi mi tylko o zasygnalizowanie pewnego schematu. To w jaki sposób rozwinąć i wzbogacić podsunięte przeze mnie rozwiązania, zostawiam waszej inwencji twórczej.

Poniżej zamieszczone są jedynie wydruki, natomiast wszystkie trzy przykłady znajdziecie na dysku Public Domain Amigowca.

Przykład 1

00 :	E-3	01000	G-2	03000	G-2	06000	---	00000
01 :	E-3	01C10	---	00000	---	00A0A	---	00000
02 :	G-3	05000	---	00000	G-2	06A0A	---	00000
03 :	G-3	05C0A	---	00000	---	00000	---	00000
04 :	E-3	02A0A	---	00000	G-2	06C3A	---	00000
05 :	---	00000	---	00000	---	00488	---	00000
06 :	G-3	05000	---	00000	---	00400	---	00000
07 :	E-3	02A0A	---	00000	---	00400	---	00000
08 :	E-3	01000	---	00000	A#2	06C20	---	00000
09 :	E-3	02A0F	---	00000	---	00AA0	---	00000
10 :	G-3	05A0A	---	00000	---	00C0A	---	00000
11 :	G-3	05C0A	---	00000	D-3	06000	---	00000
12 :	E-3	02A0A	---	00000	---	00000	---	00000
13 :	---	00000	---	00000	---	00C0A	---	00000
14 :	G-3	05000	---	00000	G-2	06A06	---	00000
15 :	E-3	02A0A	---	00000	---	00C0A	---	00000
16 :	E-3	01000	A#2	04000	A#2	06000	---	00000
17 :	E-3	01C10	---	00000	---	00A0A	---	00000
18 :	G-3	05000	---	00000	A#2	06A0A	---	00000
19 :	G-3	05C0A	---	00000	---	00000	---	00000
20 :	E-3	02A0A	---	00000	A#2	06000	---	00000
21 :	---	00000	---	00000	---	00488	---	00000
22 :	G-3	05000	---	00000	---	00400	---	00000
23 :	E-3	02A0A	---	00000	---	00400	---	00000
24 :	E-3	01000	---	00000	F-3	06A06	---	00000
25 :	E-3	02A0F	---	00000	---	00A02	---	00000
26 :	G-3	05A0A	---	00000	---	00A02	---	00000
27 :	G-3	05C0A	---	00000	D-3	06000	---	00000
28 :	E-3	02A0A	---	00000	---	00488	---	00000
29 :	---	00000	---	00000	---	00C0A	---	00000
30 :	G-3	05000	---	00000	A#2	06A07	---	00000
31 :	E-3	02A0A	---	00000	---	00000	---	00000
32 :	E-3	01000	C-3	03000	C-3	06000	---	00000
33 :	E-3	01C10	---	00000	---	00A0A	---	00000
34 :	G-3	05000	---	00000	C-3	06A0A	---	00000
35 :	G-3	05C0A	---	00000	---	00000	---	00000
36 :	E-3	02A0A	---	00000	C-3	06000	---	00000
37 :	---	00000	---	00000	---	00488	---	00000
38 :	G-3	05000	---	00000	---	00400	---	00000
39 :	E-3	02A0A	---	00000	---	00000	---	00000
40 :	E-3	01000	---	00000	A#2	06000	---	00000
41 :	E-3	02A0F	---	00000	---	00000	---	00000
42 :	G-3	05A0A	---	00000	G-2	06000	---	00000
43 :	G-3	05C0A	---	00000	---	00000	---	00000
44 :	E-3	02A0A	---	00000	D#3	06000	---	00000
45 :	---	00000	---	00000	---	00C2A	---	00000



46	G-3	05000	---	00000	G-2	06A07	---	00000
47	E-3	02A0A	---	00000	---	00000	---	00000
48	E-3	01000	D#3	04000	A#2	06000	---	00000
49	E-3	01C10	---	00000	---	00A0A	---	00000
50	G-3	05000	---	00000	---	00A0A	---	00000
51	G-3	05C0A	---	00000	A#2	06000	---	00000
52	E-3	02A0A	---	00000	---	00C0A	---	00000
53	---	00000	---	00000	---	00000	---	00000
54	G-3	05000	---	00000	A#2	06C20	---	00000
55	E-3	02A0A	---	00000	---	00000	---	00000
56	E-3	01000	---	00000	---	00000	---	00000
57	E-3	02A0A	---	00000	---	00000	---	00000
58	G-3	05000	---	00000	D#3	06C10	---	00000
59	G-3	05C0A	---	00000	---	00000	---	00000
60	E-3	02A0A	---	00000	A-2	06102	---	00000

Przykład oparty jest na schemacie G-moll - A#-dur - C-moll - D#-dur i podobnie rzecz ma się z linią basu, która została stworzona wyłącznie z dźwięków składowych poszczególnych akordów. W tym przykładzie wiodącą rolę odgrywa perkusja, a bas jedynie z nią współgra. Jest to częste i dość wygodne rozwiązania, ponieważ partią perkusji maskujemy monotonię bądź brak koncepcji w linii basu.

Przykład 2

00	F-3	01000	A#2	03000	A#2	06F06	---	00000
01	---	00000	---	00000	---	00000	---	00000
02	A-3	05000	---	00000	A#2	06C10	---	00000
03	F-3	01000	---	00000	---	00000	---	00000
04	F-3	02000	A#2	03C10	A-2	06102	---	00000
05	---	00000	---	00000	A#2	063A0	---	00000
06	F-3	01000	G#2	04000	G#2	06000	---	00000
07	---	00000	---	00000	---	00000	---	00000
08	A-3	05C0A	---	00000	G#2	06C0A	---	00000
09	---	00000	---	00000	---	00000	---	00000
10	F-3	01000	---	00000	D#3	06000	---	00000
11	---	00000	---	00000	C-3	06C0A	---	00000
12	F-3	02000	---	00000	---	00000	---	00000
13	---	00000	---	00000	G#3	06000	---	00000
14	A-3	05A0A	G#2	04C0A	---	00000	---	00000
15	---	00000	---	00000	---	00000	---	00000
16	F-3	01000	F-2	03000	C-3	06000	---	00000
17	---	00000	---	00000	---	00000	---	00000
18	A-3	05000	---	00000	C-3	06C0A	---	00000
19	F-3	01000	---	00000	C-3	06000	---	00000
20	F-3	02000	F-2	03C10	---	00000	---	00000
21	---	00000	---	00000	---	00000	---	00000
22	F-3	01000	F#2	04000	C#3	06000	---	00000
23	---	00000	---	00000	---	00000	---	00000
24	A-3	05C0A	---	00000	---	00000	---	00000
25	---	00000	---	00000	---	00000	---	00000
26	F-3	01000	---	00000	C#3	06C0A	---	00000
27	---	00000	---	00000	---	00000	---	00000
28	F-3	02000	---	00000	F-3	06101	---	00000
29	---	00000	---	00000	F#3	063A0	---	00000
30	A-3	05A0A	---	00000	C#3	06320	---	00000
31	---	00000	---	00000	---	00202	---	00000
32	F-3	01000	A#2	03000	A#2	06000	---	00000
33	---	00000	---	00000	---	00000	---	00000
34	A-3	05000	---	00000	A#2	06C10	---	00000
35	F-3	01000	---	00000	---	00000	---	00000
36	F-3	02000	A#2	03C10	A-2	06102	---	00000
37	---	00000	---	00000	A#2	063A0	---	00000
38	F-3	01000	G#2	04000	G#2	06000	---	00000
39	---	00000	---	00000	---	00000	---	00000
40	A-3	05C0A	---	00000	G#2	06C0A	---	00000
41	---	00000	---	00000	---	00000	---	00000
42	F-3	01000	---	00000	D#3	06000	---	00000
43	---	00000	---	00000	C-3	06C0A	---	00000
44	F-3	02000	---	00000	---	00000	---	00000
45	---	00000	---	00000	G#3	06000	---	00000
46	A-3	05A0A	G#2	04C0A	---	00000	---	00000
47	---	00000	---	00000	---	00000	---	00000
48	F-3	01000	F-2	03000	C-3	06000	---	00000
49	---	00000	---	00000	---	00000	---	00000
50	A-3	05000	---	00000	---	00000	---	00000
51	F-3	01000	---	00000	C-3	06000	---	00000
52	F-3	02000	F-2	03C10	---	00000	---	00000
53	---	00000	---	00000	---	00000	---	00000
54	F-3	01000	F#2	04000	C#3	06000	---	00000
55	---	00000	---	00000	---	00000	---	00000
56	A-3	05C0A	---	00000	---	00000	---	00000
57	---	00000	---	00000	---	00000	---	00000
58	F-3	01000	---	00000	C#3	06C0A	---	00000
59	---	00000	---	00000	---	00000	---	00000
60	F-3	02000	---	00000	F-3	06101	---	00000
61	---	00000	---	00000	F#3	063A0	---	00000
62	A-3	05A0A	---	00000	C#3	06320	---	00000
63	---	00000	---	00000	---	00202	---	00000

Schemat A#-moll - G#-dur - F-moll - F#-dur (czyli też nic oryginalnego). Tutaj ważniejszą rolę odgrywa bas, który towarzyszy zmianom akordów akcentując je. Partia basu (tak jak w poprzednim przykładzie) została wykreowana z dźwięków składowych odpowiednich akordów. Z wyjątkiem ślizgów z wykorzystaniem komendy 3xx. W tym przypadku dźwięk niższy (zazwyczaj o pół tonu) "dojeżdża" (komenda 1xx) do dźwięku składowego.

Przykład 3

00	F-3	01000	C-3	03000	C-3	06000	C-1	07F06
01	---	00000	---	00000	---	00000	---	00000
02	---	00000	---	00000	---	00000	---	00000
03	---	00000	---	00000	---	00000	---	00000
04	G-3	02000	---	00000	C-3	06C0A	D#2	07000
05	---	00000	---	00000	---	00000	---	00000
06	F-3	01000	---	00000	C-3	06000	C-2	07C0A
07	---	00000	---	00000	---	00000	---	00000
08	---	00000	---	00000	---	00000	C-2	07000
09	---	00000	---	00000	---	00000	---	00000
10	F-3	01000	---	00000	C-3	06C0A	D#2	07000
11	---	00000	---	00000	---	00000	---	00000
12	G-3	02000	---	00000	F#2	06102	C-2	07C0A
13	---	00000	---	00000	G-2	063A0	---	00000
14	F-3	01000	---	00000	C-3	06000	D#2	07000
15	---	00000	---	00000	---	00000	---	00000
16	F-3	01000	A#2	04000	A#2	06000	---	00000
17	---	00000	---	00000	---	00000	---	00000
18	---	00000	---	00000	---	00000	A#1	07C10
19	---	00000	---	00000	F-3	06000	---	00000
20	G-3	02000	---	00000	---	00000	D-2	07000
21	---	00000	---	00000	---	00000	---	00000
22	F-3	01000	---	00000	D-3	06000	---	00000
23	---	00000	---	00000	---	00000	---	00000
24	---	00000	---	00000	F-3	06000	F-2	07C20
25	---	00000	---	00000	---	00000	---	00000
26	F-3	01000	---	00000	---	00000	A#1	07000
27	---	00000	---	00000	D-3	06000	---	00000
28	G-3	02000	---	00000	---	00000	---	00000
29	---	00000	---	00000	---	00000	---	00000



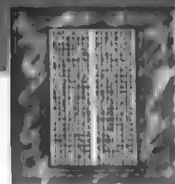
	X	Xm	X°	X5+	X4(sus)	X7	Xm7	X7+	X6	X7-5	X9	X9-
C	C-E-G	C-Es-G	C-Es-Ges-(A°)	C-E-Gis	C-F-G	C-E-G-B	C-Es-G-B	C-E-G-H	C-E-G-A	C-E-Ges-B	C-E-G-B-D	C-E-G-B-Des
Cis(Des)	Cis-Eis-Gis	Cis-E-Gis	Cis-E-G-(B)	Cis-Fis-A°	Cis-Fis-Gis	Cis-Eis-Gis-H	Cis-E-Gis-H	Cis-Eis-Gis-His	Cis-Eis-Gis-Ais	Cis-Eis-G-H	Cis-Eis-Gis-H-Dis	Cis-Eis-Gis-H-D
D	D-Fis-A	D-F-A	D-F-As-(Ces)	D-Fis-Ais	D-G-A	D-Fis-A-C	D-F-A-C	D-Fis-A-Cis	D-Fis-A-H	D-Fis-As-C	D-Fis-A-C-E	D-Fis-A-C-Es
Es(Dis)	Es-G-B	Es-Ges-B	Es-Ges-A°-(C°)	Es-G-H	Es-As-B	Es-G-B-Des	Es-Ges-B-Des	Es-G-B-D	Es-G-B-C	Es-G-A°-Des	Es-G-B-Des-F	Es-G-B-Des-Fes
E	E-Gis-H	E-G-H	E-G-B-(Des)	E-Gis-His	E-A-H	E-Gis-H-D	E-G-H-D	E-Gis-H-Dis	E-Gis-H-Cis	E-Gis-B-D	E-Gis-H-D-Fis	E-Gis-H-D-F
F	F-A-C	F-As-C	F-As-Ces-(D°)	F-A-Cis	F-B-C	F-A-C-Es	F-As-C-Es	F-A-C-E	F-A-C-D	F-A-Ces-Es	F-A-C-Es-G	F-A-C-Es-Ges
Fis(Ges)	Fis-Ais-Cis	Fis-A-Cis	Fis-A-C-(Es)	Fis-Ais-D°	Fis-H-Cis	Fis-Ais-Cis-E	Fis-A-Cis-E	Fis-Ais-Cis-Eis	Fis-Ais-Cis-Dis	Fis-Ais-C-E	Fis-Ais-Cis-E-Gis	Fis-Ais-Cis-E-G
G	G-H-D	G-B-D	G-B-Des-(Fes)	G-H-Dis	G-C-D	G-H-D-F	G-B-D-F	G-H-D-Fis	G-H-D-E	G-H-Des-F	G-H-D-F-A	G-H-D-F-As
As(Gis)	As-C-Es	As-Ces-Es	As-Ces-D°-(F°)	As-C-E	As-Des-Es	As-C-Es-Ges	As-Ces-Es-Ges	As-C-Es-G	As-C-Es-F	As-C-D°-Ges	As-C-Es-Ges-B	As-C-Es-Ges-A°
A	A-Cis-E	A-C-E	A-C-Es-(Ges)	A-Cis-Eis	A-D-E	A-Cis-E-G	A-C-E-G	A-Cis-E-Gis	A-Cis-E-Fis	A-Cis-Es-G	A-Cis-E-G-H	A-Cis-E-G-B
B(Ais)	B-D-F	B-Des-F	B-Des-Fes-(G)	B-D-Fis	B-Es-F	B-D-F-As	B-Des-F-As	B-D-F-A	B-D-F-G	B-D-Fes-As	B-D-F-As-C	B-D-F-As-Ces
H	H-Dis-Fis	H-D-Fis	H-D-F-(As)	H-Dis-G	H-E-Fis	H-Dis-Fis-A	H-D-Fis-A	H-Dis-Fis-Ais	H-Dis-Fis-Gis	H-Dis-F-A	H-Dis-Fis-A-Cis	H-Dis-Fis-A-C

30	F-3 01000	--- 00000	A#2 06000	G-1 07000
31	--- 00000	--- 00000	--- 00000	--- 00000
32	F-3 01000	G-2 03000	G-2 06000	--- 00000
33	--- 00000	--- 00000	--- 00000	--- 00000
34	--- 00000	--- 00000	--- 00000	G-1 07C0A
35	--- 00000	--- 00000	--- 00000	--- 00000
36	G-3 02000	--- 00000	--- 00000	G-1 07000
37	--- 00000	--- 00000	--- 00000	--- 00000
38	F-3 01000	--- 00000	G-2 06000	--- 00000
39	--- 00000	--- 00000	--- 00000	--- 00000
40	--- 00000	--- 00000	--- 00000	D-2 07C10
41	--- 00000	--- 00000	--- 00000	--- 00000
42	F-3 01000	--- 00000	--- 00000	A#1 07000
43	--- 00000	--- 00000	--- 00000	--- 00000
44	G-3 02000	--- 00000	G-2 06000	--- 00000
45	--- 00000	--- 00000	--- 00000	--- 00000
46	F-3 01000	--- 00000	G-2 06C0A	D-2 07C10
47	--- 00000	--- 00000	--- 00000	--- 00000
48	F-3 01000	G#2 04000	G#2 06000	C-2 07000
49	--- 00000	--- 00000	--- 00000	--- 00000
50	--- 00000	--- 00000	G#2 06C10	--- 00000
51	--- 00000	--- 00000	--- 00000	--- 00000
52	G-3 02000	--- 00000	D#3 06000	--- 00000
53	--- 00000	--- 00000	--- 00000	--- 00000
54	F-3 01000	--- 00000	C-3 06000	C-2 07000
55	--- 00000	--- 00000	--- 00000	--- 00000
56	--- 00000	A#2 04000	A#2 06000	--- 00000
57	--- 00000	--- 00000	--- 00000	--- 00000
58	F-3 01000	--- 00000	A#2 06C0A	--- 00000
59	--- 00000	--- 00000	--- 00000	--- 00000
60	G-3 02000	--- 00000	F-3 06000	D-2 07000
61	--- 00000	--- 00000	--- 00000	--- 00000
62	F-3 01000	--- 00000	D-3 06000	--- 00000
63	--- 00000	--- 00000	--- 00000	--- 00000

Oparty na najpopularniejszej chyba zmianie akordów C-moll - A#-dur - G-moll - G#-dur - A#-dur. Oprócz basu występuje tu także partia pianina, również oparta na dźwiękach składowych. Próba "wariacji" na basie w pierwszej połowie patternu pokazuje, że układanie akompaniamentu w taki sposób wcale nie musi być monotonne i pozbawione dynamiki. Można próbować zmieniać oktawy, tworzyć pogłos lub wprowadzić drugi instrument basowy (np. klangująca gitara basowa). Możliwości jest bardzo wiele.



To wszystko w tym odcinku. Gorąco zachęcam do eksperymentów. Aż do utraty słuchu. Przypominam, że przykłady znajdą się na dysku PD. Zatem wszystkim tym, którym nie uda się skompletować instrumentów, aby prześledzić zaprezentowane przykłady, radzę zajrzeć na dysk PD. Zresztą nie tylko po to. ❑



PRIMA C-PRILIS

Tytuł, jak i treść artykułu nawiązuje oczywiście do 1 kwietnia, [redacted] więc odwiecznego święta wszelkiego typu wygłupów.

Dusza psotnika drzemie chyba w każdym z nas, tylko nie wszyscy chcą się do tego przyznać. Jednak szczególnym zamiłowaniem do "psot" charakteryzują się niektórzy - pożal się Boże - programiści, czy raczej - koderzy. To ich dziełem są wymyślne wirusy, które lepiej lub gorzej radzą sobie [redacted] formatowaniem naszych dysków, zawieszaniem komputera, itp. Nie jest to dobry sposób [redacted] rozładowanie swej inwencji twórczej, więc zostawmy go dla sfrustrowanych osobników, którzy rozładowują w ten sposób swe kompleksy.

Prima aprilis jednak do czegoś zobowiązuje, więc proponuję napisanie [redacted] programiku. Przy okazji zobaczymy, jak podejść do rozwiązania ja[redacted] problemu. Przejdźmy do rzeczy.

Najpierw sformułujemy problem: chcemy zrobić komuś numer. Numer [redacted] zostać wykonany na komputerze. Nasze założenia, to całkowita nieszkodliwość zabawy i powrót do stanu wyjściowego sprzed rozpoczęcia działania. Opracujmy zarys rozwiązania. Nie jest naszą intencją pisanie wirusów, czyli [redacted] nie może się powielać (pamiętajmy założenie: powrót do stanu [redacted] zabawy). Powstaje problem uruchomienia dzieła w krytycznym dniu. Możliwe są dwa rozwiązania. Pierwsze, to bezpośrednie uruchomienie programu [redacted] pracującym komputerze, jednak do tego potrzebne jest intymne spotkanie w cztery oczy (dyski?, piksele?) z komputerem, bez świadków. Drugie, [redacted] zapisanie programu pod jakąś nazwą na twardy dysk i wstawienie do sekwencji startowej krótkiej instrukcji wywołania. To drugie rozwiązanie wiąże się jednak z ingerencją [redacted] zawartość twardego dysku, stąd może znaleźć [redacted] stosowanie tylko [redacted] przypadku wyjątkowo dobrze znanych i wyrozumiałych ofiar (pamiętajmy o późniejszym posprzątaniu!). My stosując założenie o nieszkodliwości przyjmujemy rozwiązanie pierwsze mając przy tym nadzieję, że ofiara to osoba gościnna, która zrobi nam w kuchni herbatę, nierozmyślnie przekazując swój komputer w bardziej doświadczone ręce.

"JAK?" już wiemy, teraz pytanie: CO??? Co można zrobić w komputerze by nie było to groźne? Można np. pobłyskać ekranem (od wersji systemu operacyjnego 2.0 błyskom towarzyszy zdefiniowany w preferencjach efekt dźwiękowy - można go zmienić, lecz znów należy wziąć poprawkę na charakter ofiary). Można zacząć opuszczać powoli aktualny ekran. Jeśli jest otwartych kilka ekranów, można je poprzekładać kilka razy, wracając oczywiście na koniec [redacted] ekran początkowy. Można wreszcie przesunąć cały widok na ekranie w którąkolwiek [redacted] stron (i wrócić do punktu początkowego). Mierne efekty spadających literek [redacted] pecetach w przypadku Amigi można zastąpić spadającymi ekranami - a co, nie będziem poprzestawać na małym! Zresztą pomysłów jest dużo, jeśli np. ofiara pracuje na oknach zawsze mogą [redacted] zacząć się powoli przemieszczać itp.

Jak widzimy wszystkie wyżej wymienione operacje ograniczają się do operowania na systemowych procedurach i ich użycie nie powinno grozić jakimiś daleko idącymi konsekwencjami. Tu jednak dochodzimy do ograniczeń. Otóż najlepsze efekty dadzą się zaobserwować wtedy, gdy będzie otwartych kilka ekranów czy okien. Stąd naszym zadaniem jest zmuszenie ofiary do odpalenia kilku programów (np. "Przyniosłem kilka fajnych procedurek. Zgraj je [redacted] sobie na twardziela (Directory Opusem, rzecz jasna), [redacted] potem pooglądamy moje wstawki graficzne (Deluxe Paint) i zredagujemy list do Amigowca pt. "Więcej o C (i CED w pamięci). W przypadku, gdy komputer jest wyposażony w karty graficzne, trzeba uważać - lepiej wtedy wcześniej spokojnie na czystej pamięci sprawdzić nieszkodliwość programu. Również [redacted] monitorach pracujących z częstotliwością odchylania poziomego od 15 kHz do 30 kHz i więcej należy sprawdzić, jak zachowują się one podczas opuszczania ekranów.

Mamy zdefinowany problem, zarys rozwiązania, ewentualne ograniczenia i błędy. Przejdźmy do opracowania technicznej strony zagadnienia. Czym? Tu odpowiedzi [redacted] proste. Błyskaniem ekranami zajmuje się funkcja:

void DisplayBeep(struct Screen *screen)

której argumentem jest wskaźnik [redacted] ekran, na którym [redacted] być widoczny błysk. Jeśli podamy wartość NULL, wtedy błysk jest widoczny na wszystkich ekranach. Przesuwanie ekranu (do wersji systemu 2.0 tylko w pionie) realizuje in[redacted] funkcja:

void MoveScreen(struct Screen *screen, LONG dx, LONG dy)

gdzie [redacted] to wskaźnik na przesuwany ekran, [redacted] dx i dy to wartości przesunięcia odpowiednio w poziomie i [redacted] pionie wyrażone [redacted] pikselach. Przydatną będzie też funkcja:

void ScreenToFront(struct Screen *screen)

która przesuwa do przodu ekran wskazywany przez wskaźnik screen. Wskaźnik [redacted] aktywny ekran znajdziemy w składowej ActiveScreen struktury IntuitionBase. Natomiast wskaźnik [redacted] pierwszy [redacted] ekranów znajdujących się w pamięci zawiera składowa FirstScreen tej samej struktury. Ponieważ jest to wskaźnik na strukturę Screen, więc następny ekran będzie wskazywany przez składową NextScreen tej struktury. Ostatni element listy ma jak zwykle wartość NextScreen równą NULL. Niewiele większy problem pozostaje [redacted] przesuwaniem widoku [redacted] bokl. [redacted] tym celu należy zmienić składowe DxOffset i DyOffset struktury ViewPort, która jest składową struktury Screen (UWAGA! Fragmentem struktury Screen jest struktura Viewport, nie jest to wskaźnik [redacted] ViewPort!). Zmiana tych parametrów nic jeszcze nie powoduje, bowiem copper musi zostać poinformowany o nowej sytuacji. Wykonują to dwie syste[redacted] procedury:

void MakeScreen(struct Screen *screen)

zawierająca wskaźnik na uaktualniany ekran oraz:

void RethinkDisplay()

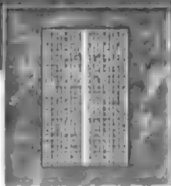
Przykładowe rozwiązanie pokazuje dołączony listing. Przed ostatecznym skompilowaniem warto dobrać czasy opóźnień, po jakich mają się pojawić efekty - w załączonym programie jest dodatkowo pewien pseudolosowy rozrzut w czasie. Wszystkie efekty NALEŻY sprawdzić [redacted] konkretnym komputerze, by uniknąć niepotrzebnych rumieńców i tłumaczeń. Ja przetesowałem działanie nie tylko [redacted] komputerze, [redacted] także na osobie - tu ukłony w stronę miłego składacza (a raczej składaczki) tego artykułu. Proszę [redacted] wybaczenie, to było tylko dla dobra czytelników!

Pamiętajmy, że przy tego typu żartach na komputerze obowiązuje podobna zasada jak w medycynie: po pierwsze - NIE SZKODZIĆ! I na dodatek 1 kwietnia trwa tylko jeden dzień.

Do zobaczenia.

```
#include <proto/istuition.h>
#include <proto/graphics.h>
#include <proto/exec.h>
#include <proto/dos.h>
#include <intuition/intuitionbase.h>
#include <graphics/gfxbase.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/* Wskaźniki na używane biblioteki. */
struct IntuitionBase *IntuitionBase;
struct GfxBase *GfxBase;
extern struct DosLibrary *DOSBase;
```



```
/* Deklaracje funkcji. */

void CleanUp( STRPTR message );


void main()
   {
      UBYTE loop, koniec;
      UWORD loopik, suma;
      struct Screen *ekran, *pierwszy;

      /* Otwarcie bibliotek. */

      IntuitionBase = (struct IntuitionBase *) OpenLibrary
                                          ( "intuition.library", 0 );

      if( !IntuitionBase )
         CleanUp( (STRPTR) "Nie mogę otworzyć intuition.library!" );

      GfxBase = (struct GfxBase *) OpenLibrary( "graphics.library", 0 );

      if( !GfxBase )
         CleanUp( (STRPTR) "Nie mogę otworzyć graphics.library!" );

      /* Wykonanie pięciu różnych "efektów specjalnych". */

      for ( koniec=0; koniec<5; koniec++ )
         {
            /* Mała zwłoka. */

            Delay( 50 * ( 200 + rand() % 200 ) );

            /* Efekty. */

            switch ( koniec )
               {
                  case 0:
                     /* Błyski ekranów. */

                     for ( loop=0; loop<100; loop++ )
                        {
                           DisplayBeep(NULL);
                           Delay(5);            /* 1/10 sekundy zwłoki. */
                        }

                     break;

                  case 1:
                     /* Opuszczenie aktywnego ekranu. */

                     for ( loop=1; loop<40; loop++ )
                        {
                           MoveScreen( IntuitionBase->ActiveScreen, 0, 1 );
                           Delay(100);          /* 2 sekundy zwłoki. */
                        }

                     break;

                  case 2:
                     /* Przeskok ekranów. */

                     pierwszy = IntuitionBase->ActiveScreen;

                     /* Pierwotny ekran. Wypada go potem odtworzyć. */

                     for ( loop=0; loop<50; loop++ )
                        {
                           ekran = IntuitionBase->FirstScreen;

                           while ( ekran->NextScreen )
                           ekran = ekran->NextScreen;

                           ScreenToFront( ekran );
                           Delay(10);
                        }

                     /* Pierwotny ekran ponownie na widok. */

                     ScreenToFront( pierwszy );
                     break;

                  case 3:
                     /* Odlot widoku. */

                     /* Ucieczka widoku. */

                     for ( loopik=0; loopik<200; loopik++ )
                        {
                           IntuitionBase->ActiveScreen->ViewPort.
                                                            DxOffset++;
                           IntuitionBase->ActiveScreen->ViewPort.
                                                            DyOffset--;
                           MakeScreen( IntuitionBase->ActiveScreen );
                           RethinkDisplay();
                        }

                     /* Powrót widoku. */

                     for ( loopik=0; loopik<200; loopik++ )
                        {
                           IntuitionBase->ActiveScreen->ViewPort.
                                                            DxOffset--;
                           IntuitionBase->ActiveScreen->ViewPort.
                                                            DyOffset++;
                           MakeScreen( IntuitionBase->ActiveScreen );
                           RethinkDisplay();
                        }

                     break;

                  case 4:
                     /* Spadające ekrany. */

                     ekran = IntuitionBase->FirstScreen;
                                             /* Pierwszy ekran listy. */

                     while(1)
                        {
                           for ( loopik = 1, suma = 0; suma += loopik,
                                       suma <ekran->Height; loopik++ )
                              MoveScreen( ekran, 0, loopik );

                           if ( !( ekran = ekran->NextScreen ) )
                              break;
                        }

                     break;
               }
         }

      CleanUp( (STRPTR) "I TO BY BYŁO NA TYLE..." );
   }

/* Funkcja zwalnia wszystko, co jest do zwolnienia. */

void CleanUp( STRPTR message )
{
   /* Zamknięcie bibliotek. */

   if( GfxBase )
      CloseLibrary( (struct Library *) GfxBase );

   if( IntuitionBase )
      CloseLibrary( (struct Library *) IntuitionBase );

   printf( "%s", message );
   exit(0);
}
```



FG 24

Nowość !!!

FG 24 - umożliwia wczytywanie w czasie rzeczywistym kolorowego obrazu telewizyjnego (system PAL) z magnetowidu, kamery lub tunera TV-SAT w trybie 24 bitowym

FG 24 - współpracuje poprzez złącze CENTRONICS z każdym typem AMIGI (KICKSTART 2.0 lub nowszy, 1,5 MB pamięci - wkrótce wersja dla systemu 1,3)

CENA 3.300.000 zł



Zdjęcia wczytane przy pomocy FG 24

Akcesoria do komputerów AMIGA

[illegible]

Mega RAM 2 MB pamięci	[illegible]
Mega RAM 0 MB pamięci	[illegible]
Mega RAM HD 2 MB pamięci	[illegible]
Mega RAM HD 0 MB pamięci	[illegible]
AT BUS HD - kontroler dysku twardego	1.690.000.-

[illegible]

Sampler Mono (40 kHz)	350.000.-
Sampler Stereo (30 kHz)	475.000.-
Midi interface (1 IN, 1 OUT, 1 THRU)	350.000.-

[illegible]

Interface do dwóch zwykłych joysticków i myszki	350.000.-

Amiga 1200

2,5" lub 3,5" Hard Disk Kit - twardy dysk w środku komputera za niższą cenę (kable łączące, instrukcja i oprogramowanie w języku polskim)	390.000.-
E1204 - 0 MB RAM, bez koprocesora	2.990.000.-

podane ceny zawierają 22% VAT

Sprzedaż za zaliczeniem pocztowym po doliczeniu kosztów przesyłki. Dla odbiorców hurtowych rabat do 40%.

Możliwość zakupu na raty!

E1204

Rozszerzenie to przeznaczone jest dla komputera **AMIGA 1200**. Powiększa pamięć komputera o **4MB** prawdziwego 32 bitowego **FAST RAM** - u.

* szybkość działania komputera **2.19** raza większa
* zegar czasu rzeczywistego pozwala na oznaczanie zbiorów właściwą datą i czasem.
* nie blokuje współpracy z kartami **PCMCIA**.
* typowe moduły pamięci **SIMM 256 kB** (1MB pamięci) lub **1 MB** (4MB pamięci), 8 lub 9 [illegible]



Dystrybutorzy naszych produktów:

AMIGA s.c.	Warszawa ul. Batorego 10,	tel.: 25-60-31 w 103
AMIGO	Wrocław ul. Leszczyńskiego,	tel.: 44-41-63
AMI-GO!	Opole ul. Kośnego 43,	tel.: 315-55
AMIKOM	Białystok ul. Piłsudskiego 38,	tel.: 436-028
bajPCtek	Kraków ul. [illegible] 8,	tel.: 22-59-72
	Nowy Sącz ul. Wąsowiczów 10,	
	Tarnów ul. [illegible] 2,	tel.: 22-40-70
ALL IN ONE	Łódź ul. Piotrkowska 111,	tel.: 327-324
CONTRA D.H. BIM	Olsztyn, ul. Piłsudskiego 46,	tel.: 26-72-13
IMEX	Gorzów Wlkp. ul. Sczanieckiej 41,	tel.: 74-115
INFO	Legnica ul. Złotoryjska	tel.: 56-06-55
JTT Computer [illegible]	Wrocław, ul. Świdnicka 19,	tel.: 44-12-33
[illegible] s.c.	Katowice Pl. Rostka 3,	tel.: 515-132
	Rybnik ul. Wiejska 19,	tel.: 233-56
	Bielsko - Biała Pl. Wolności 3,	tel.: 229-70
[illegible]	Raszyn ul. Mickiewicza 14,	tel.: 56-08-91
VADIM	Zielona [illegible] ul. Kupiecka 1,	tel.: 656-72
XYZ - Mikrokomputery s.c.	Lublin ul. Okopowa 6	tel.: 21-394

ELSAT s.c.
00-714 Warszawa
ul. Czerniakowska 28 B
tel.: (22) 40-58-76
(2) 642-96-05
fax: (22) 41-52-80

ELSAT

POTĘGA MOŻLIWOŚCI - ZAUFAJ DOŚWIADCZENIU !



GVP to najlepszy wybór dla posiadacza każdej Amigi

Firma Great Valley Products jest liderem w dziedzinie peryferii i rozszerzeń do Amigi od 1988 roku.



TYLKO RZECZY NAJLEPSZE SĄ WYSTARCZAJĄCO DOBRE !
KUPUJESZ DROŻEJ - WYCHODZI TANIEJ !



A4000 G-Force [illegible]
Jeśli jesteś profesjonalistą i czujesz się skrępowany niewystarczającą prędkością Twojej Amigi 4000 lub 3000, [illegible] ta sprawi, że zapomnisz o tym problemie na całe lata. A4000 [illegible] 040 to m.in. zegar 40MHz, do 128MB 32-bitowej bardzo szybkiej pamięci, gniazdo rozszerzeń na przykład dla opcjonalnego modułu SCSI-II FAST. [illegible] pracuje z oboma typami [illegible], A4000T (z procesorami 030 lub 040)

G-FORCE



G-Force '040/33 Combo
Karta Combo rozbudowana do absolutnego maksimum. Twoje programy będą kunać [illegible] olbrzymiej mocy pracującego z częstotliwością 33MHz procesora 68040 [illegible] mocy dostarczyć do 64MB 32-bitowej pamięci.
Oczywiście nasz interfejs SCSI II znajduje się na karcie, dodajemy także jako bonus dodatkowe porty: równoległy i buforowany, [illegible] port szeregowy.



4008 SCSI II / 4008 SCSI2FAST
Wejdź do świata SCSI przy pomocy jednej z tych prostych w instalacji kart. Natychmiast [illegible] dostęp do tysięcy urządzeń, takich jak twarde dyski, napędy SyQuest, CD-ROM. Możesz dołączyć do swojej Amigi 2000 do siedmiu takich urządzeń. [illegible] pracuje także z [illegible] 2000 i umożliwia rozszerzenie pamięci RAM o 8MB. Karta [illegible] wyposażona jest w [illegible] układ, pozwalający na osiągnięcie rewelacyjnej maksymalnej szybkości przesyłu do 10MB/s, a dzięki miejscu na rzędem modułów SIMM, możliwe jest rozszerzenie pamięci do 256MB. Dzięki niewielkim rozmiarom karty (technologia montażu powierzchniowego), możliwe jest zamontowanie dowolnego 3.5" [illegible] bezpośrednio na karcie, zapewniając [illegible] wygodę. Przekonaj się o różnicy.



IV-24 [illegible] -IMPACT VISION 24- Profesjonalny Genlock
To jest to, czego szukałeś, potrzebując profesjonalnej jakości genlocka do Twojej Amigi 2000, 3000 lub 4000. Wysoko zintegrowane układy dają najlepszy, najczystszy obraz. Przy możliwości pracy na wyjściu i wejściu z sygnałami RGB, composite video, SVHS, component (Betacam oraz M-II), przy 24-bitowym frame buferze i pracującym w czasie rzeczywistym digitizerze obrazu, IV24 jest genlockiem potrzebnym każdemu profesjonaliście.
Dodawane oprogramowanie uzupełnia obraz Profesjonalnego Genlocka.

TBC Plus



TBC Plus
Ten profesjonalny, cyfrowy Time Base Corrector (Korektor Bazy Czasowej) pracuje w najwyższej jakości, standardzie 4:2:2 (8 bitów).
PLUS: karta zawiera także pracujący w czasie rzeczywistym frame-grabber oraz 24-bitowy frame-buffer. PLUS: karta potrafi odbierać i [illegible]ować kody czasowe SMPTE/EBU. PLUS: to [illegible] urządzenie transkoduje sygnały PAL-NTSC-SECAM, zarówno composite video jak i Y/C.
PLUS: możliwość wyboru jednego z trzech kanałów video (composite video i Y/C).
PLUS: programowalne efekty specjalne.



G-Lock
Możesz łączyć w jedną całość obraz z [illegible] oraz grafiką Amigi przy pomocy G-Lock'a i dowolnej Amigi! Pozbędziesz się problemów dzięki wbudowanemu [illegible] pomiędzy Composite Video, Y/C, RGB oraz YUV. [illegible] wykorzystanie możliwości kości AGA, ale również także "zwykłych" Amig 500, 2000 i 3000. Doskonały, programowy [illegible] czyni łatwym sterowanie G-Lockiem. Użytkownicy programu Scala mają do dyspozycji moduł EX przeznaczony dla G-Lock'a, pozwalający [illegible] w ich prezentacjach. Dalej jeszcze możliwość [illegible] dwukanałowego dźwięku, a sądzie sam, że G-Lock to najlepszy wybór dla każdego użytkownika [illegible].





EGS 28/24 SPECTRUM
Zestaw w tele kości AGA, kupując ją pracującą w czasie rzeczywistym, 24-bitową kartę graficzną. Programowalne rozdzielczości z maksymalną 1600x1280 punktów! 800x600 punktów w 24 bitach! [illegible] karty dodajemy specjalny kabel przełotowy, umożliwiający pracę z jednym monitorem. Wiele programów już w tej chwili pracuje z kartą SPECTRUM, dodajemy też [illegible] doskonały program EGS Paint. [illegible] swojej Amidze możliwości graficzne stacji roboczej już dziś i patrz, co tracisz!



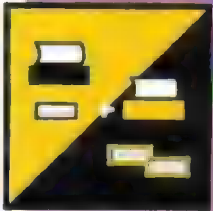
A1230 TURBO+ [illegible] II
[illegible] zegarze procesora 50MHz, opcjonalnie 40 MHz, możesz być właścicielem [illegible] Amigi 1200 na świecie! Już dziś możesz mieć do 32MB 32-bitowej pamięci. Z procesorem z zegarem 50MHz i 40MHz koprocesorem, twoje obliczenia [illegible] nigdy nie były szybsze. Prosta instalacja w złączu pod spodem A1200 nigdy nie narusza gwarancji, a przy kartach Series II zyskujesz uniwersalność naszego dodatkowego gniazda rozszerzeń. Kupując A1230 SCSI Kit możesz wyposażyć swoją A1200 w najszybszy interfejs SCSI (wyposażony w gniazdo zewnętrzne, na przykład dla CD-ROM). Po prostu podłączasz go do gniazda karty. Inne rozszerzenia to między innymi [illegible] karta dźwiękowa z samplerem i frame grabber [illegible] obrazu [illegible] pracujący w czasie rzeczywistym).





DSS8+ Po prostu genialne! To najmniej [illegible] i najbardziej profesjonalny sampler (digitizer dźwięku) dotychczas skonstruowany. Zrobiony z odpornego na uderzenia, przezroczystego tworzywa, jest to sampler do Amigi, który musisz mieć.
Dołączany program Digital Sound Studio pozwala na samplowanie, edycję, komponowanie, odtwarzanie dźwięku w stereo, jak również na zapis gotowych utworów w formacie .mod (Tracker).



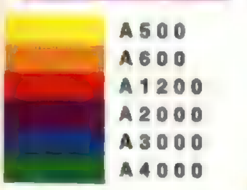


[illegible] VFX 2.0
Jeśli chcesz użyć systemu VoiceMail, naciśnij 1.
Jeśli chcesz wysłać fax, naciśnij 2.
Jeśli jeszcze chcesz, by wszystkie funkcje były [illegible] realizowane w zaprogramowanym czasie, i abyś był powiadamiany, gdy [illegible] do ciebie, kup [illegible] VFX 2.0 dziś! Pełna integracja, [illegible] nieograniczonej liczby "skrzynek pocztowych" i odbioru prywatnych faksów. Możesz tworzyć faxy z wydruków generowanych przez dowolny program. Możesz [illegible] z [illegible] miejsca i przesłać w dowolne [illegible]. Faxowanie z i bez użycia papieru.



IoExtender
Czujesz się ograniczony brakiem możliwości? Pozwól firmie GVP wyzwolić Cię przy pomocy karty ioExtender. Na jednej karcie znajdziesz dodatkowy port równoległy, pozwalający na jednoczesne podłączenie drukarki i digitizera (jak np. DSS8+). Nigdy więcej kłopotliwych zewnętrznych przełączników. Na karcie znajdują się także dwa (tak, dwa!) bardzo szybkie, buforowane porty szeregowe. Nigdy więcej straconych danych, nigdy więcej zawieszeń komputera przy szybkiej transmisji przez modem. Zwolnij porty i odzyskaj [illegible] mocy swojej Amigi kupując kartę IoExtender!

ImageFX
Doskonale zintegrowana obróbka grafiki. ImageFX to jedyny pakiet do [illegible] grafiki, jakiego będziesz kiedykolwiek potrzebował. To optymalne rozwiązanie dające nie tylko możliwość konwersji pomiędzy różnymi formatami grafiki, jak TIFF, GIF i TARGA, ale [illegible] bardzo rozbudowany, 24-bitowy program [illegible] malowania i retuszu. Efekty swojej pracy widzisz natychmiast. Wygładzanie brzegów wstawianych elementów, kanał Alpha, operacje ustawiane dla CMY/HSV/YUV/YIQ, skanowanie i digitalizacja, wybór obszaru do obróbki... Wszystko to jest w ImageFX v1.50.



aram
EXPORT - IMPORT - REKLAMA
Autoryzowany Dealer firmy GVP

ul. Śniadeckich 17
00-654 Warszawa
tel./fax (22) 29-54-99

Amiga jest nazwą zastrzeżoną Commodore-Amiga Inc.
GVP jest nazwą zastrzeżoną Great Valley Products, Inc.

Skład na komputerze AMIGA 2000 z kartą G-Force 030-50 Combo oraz EGS 28/24 SPECTRUM.
CAT-SOFT

Prowadzimy sprzedaż hurtową, detaliczną oraz wysyłkową za zaliczeniem pocztowym



Multimedialna magia dla Amigi



Producent: Media Point

Dystrybutor: Activa International Inc.
Cena: ???
Zgodność: 3.0

Podobno żaden multimedialny pakiet na Amigę nie [redacted] takich możliwości prezentacyjnych jak ten. Multimedia to oczywiście coś w rodzaju komputerowej książki o wszystkim. "Książkę" składamy my sami, a pakiet z nami współpracuje.

MediaPoint mają możliwość pełnej synchronizacji wszystkich wykonywanych czynności. Współdziała z samplerami, midi, ma możliwość odgrywania filmów "full motion" (ze specjalnych dyskietek), [redacted] także sampli z twardego dyski. Nie zabrakło też możliwości odtwarzania animacji w różnych formatach (5, 7 i [redacted] oraz CDXL działający z AGA). Ponad 150 sposobów przejścia ekranu w inny ekran zapewnia brak nudy w przerwach pomiędzy grafikami. Specjalne klawisze na ekranie, naciskane przy pomocy np. myszki pozwalają na grzebanie w strukturze prezentacji i wyciskanie z niej tego, [redacted] się da.

Tworzenie prezentacji opiera się na specjalnych skryptach w języku ScriptTalk, z pełną możliwością współpracy w ARexxie.

Łatwa obsługa programu zachęca do współpracy. Wycinanie, wstawianie, kopiowanie oraz funkcja undo doprowadziły edycję tekstu do dużej prostoty. Program umożliwia stosowanie wielokolorowych czcionek w różnych wielkościach na tle np. wygładzonych w specjalnym trybie grafik. Grafiki oczywiście mogą mieć po 16 milionów kolorów i bez problemu mogą być przetworzone na możliwości kości AGA. Program jest modularny i dzięki temu może pracować z GVP IV-24, MIDI, CDTV, odtwarzaczami dysków laserowych, magnetowidami.

Atramentówka dla każdego (StarJet SJ-48)



Producent:
Star Micronics Deutschland GmbH
Westerbachstraße 59
D-60489 Frankfurt

Dystrybutor: ABC Data, firma Baza
Cena: ok. 7 mln
Zgodność: Wszystkie systemy

Każdy posiadacz Amigi marzy o jakiejś lepszej drukarce niż zwykła igłówka. Nawet 24 igłowe [redacted] dają takich wydruków jak prawdziwe laserówki, czy drukarki atramentowe. Ostatnio na rynku pojawiła się mała drukarka atramentowa o dogodnej cenie: StarJet SJ-48.

Drukarka ta uzyskuje rozdzielczość do 360*360 punktów i robi to niezwykle cicho i sprawnie. Została wyposażona w dwa komplety polskich znaków: Mazovię i Latin-2, co umożliwia jej współpracę z różnymi edytorami tekstu (przy użyciu odpowiednich driverów). "Jeszcze łatwiej" drukarka ta współpracuje z programami typu PageStream.

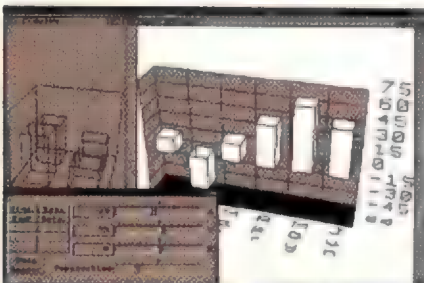
Wyposażono ją w dwie czcionki standardowe: Roman i H-Gothic o wielkościach 12, 15, 17 i 20. Czcionki te występują w trybie ekonomicznym i wysokiej jakości. Od tego w jakim trybie stworzymy nasze własne czcionki zależy ich liczba, odpowiednio 128 i 96 znaków.

Druk nie może się odbywać na papierach zbyt grubych i zbyt gładkich. Drukarka ma możliwość przepuszczenia papieru całkowicie niepogniecionego. W tym celu stawiamy ją na boku i wkładamy kartkę od tyłu. Bezproblemowy jest także wydruk adresów [redacted] kopertach lub kartkach pocztowych - można dokładnie ustalić miejsce gdzie mają być drukowane literki.

Wielkość drukarki sugeruje przeznaczenie pod laptopa, co pozwala [redacted] zaoszczędzenie miejsca, zaś cicha praca oszczędza nam uszy.

Cena drukarki nie jest zabójcza, [redacted] musimy się liczyć z tym, że cartridge z płynem do plucia kosztuje 650 tysięcy i starcza na ok. 1000 stron. Nie jest to mało, ale przy większej ilości wydruków nie jest też dużo.

Grafiki słupkowe i nie tylko



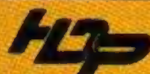
Autor: Jean-Christophe Clement
921, rang 3
St-Simon
Quebec, Canada J0H-1Y0
Internet:clemj00
@dmi.usherb.ca
Dystrybucja: Dyski PD
(np. AmigaComputing 3/94)
Zgodność: wszystkie systemy

JcGraph jest programem poszukiwanym przez wielu amigowców usiłujących przedstawić wykresy swoich badań w formie wektorowej. Do tej pory umożliwiał to jedynie program ProCalc, ale i tak nie było w nim mowy o wektorowym tekście. Grafika i owszem, ale tekst był bitmapowy. Jest to ważne, gdy nasze grafiki musimy skalować, tak by pasowały [redacted] stronie dokumentu - skalowanie grafiki bitmapowej jest mniej doskonałe niż wektorowej.

Mankamentów tych nie ma program JcGraph. Na dodatek jest on programem freeware. Stworzył go student Jean-Christophe Clement i zapowiedział, że nie będzie nowszych wersji. A szkoda. Informacje do programu wpisuje się bardzo prosto - w najzwyklejszym edytorze tekstowym wystarczy wpisać poszczególne wartości i podpisy pod kolumnami. Zakres wartości musi mieścić się pomiędzy -1000 i +1000, ale dla większości grafik jest to w zupełności wystarczające.

JcGraph ma nie tylko możliwość wyboru rodzaju grafiki, ale także jej przestrzennego umieszczenia i doboru najbardziej odpowiadających nam kolorów.

Grafiki można wyrzucać w postaci rysunków bitmapowych, a także grafik wektorowych akceptowanych przez Professional Page, czy Final Writera. Osobiście nie udało mi się wczytać wyrzucanych grafik ani do ProDrawa, ani do ArtExpression, choć ten ostatni wyraźnie akceptował podawany format. Nie też zdały się próby przerzucenia grafiki do ProDrawa z ProPage. Mimo mankamentów uważam, że warto zwrócić uwagę na ten program. Być może jest to właśnie to czego potrzebujecie. Postaram się zamieścić [redacted] [redacted] jednym z kolejnych dysków PD.



HDP Electronics s.c.

ELECTRONICS s.c.

pl. Staszica 7, 50-223 WROCŁAW, Tel. (071) 21-57-82

OFERUJE DLA KOMPUTERÓW AMIGA

DIGI LAB 2.0

System digitalizacji i obróbki obrazu w czasie rzeczywistym

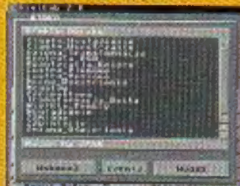
- zestaw składający się z digitalizera oraz oprogramowania DIGI LAB 2.0
- umożliwia wprowadzenie obrazu z kamery lub magnetowidu do pamięci komputera
- duża szybkość i wysoka jakość przetwarzania obrazu
- digitalizacja obrazu w czasie rzeczywistym (tryb B/W)
- prosta obsługa programu realizowana systemem okien, przycisków i suwaków
- dwie wersje programu: polska i angielska
- pełne wykorzystanie trybów graficznych Amigi (również Amigi 1200 i 4000)
- bogaty zestaw efektów specjalnych
- bogaty zestaw wejść i wyjść video w digitalizerze: VIDEO, RGB, Y/C
- naturalny sposób tworzenia animacji z bezpośredniego sygnału video
- tryb pracy 24-bitowej, umożliwiający uzyskanie 16777216 kolorów
- szczegółowa instrukcja, opisująca digitalizer i program obsługi w języku polskim



6.300.000zł



Program DIGI LAB - ekran podstawowy



Program DIGI LAB - okno efektów obrazu



Program DIGI LAB - okno animacji

Przełączniki Kickstartów

KICK-ROM 2.0P

Kickstart V2.04 dla Amigi 500/2000 790.000zł

NOWOŚĆ

KICK-ROM 1200

Kickstart V1.3 dla Amigi 1200 690.000zł

GENLOCK AX | GENLOCK PRO





4.500.000zł | 6.200.000zł

Genlocki umożliwiają mieszanie grafiki AMIGI z obrazem video.



AT-ON HD

Szybki kontroler dysku twardego AT-BUS (IDE) dla Amigi 500/500plus. Do kontrolera dołączone jest oprogramowanie do obsługi twardego dysku.

kontroler AT-ON	1.350.000zł
kontroler AT-ON+obudowa	1.700.000zł



HDP SOUND STUDIO

- REWELACYJNY PROGRAM DIGITON V1.1
- AMIGA SOUND SAMPLER
- SZCZEGÓŁOWA INSTRUKCJA OBSŁUGI W JĘZYKU POLSKIM

CENA ZESTAWU MONO	540.000zł
CENA ZESTAWU STEREO	750.000zł

SOUND SAMPLER (mono, 27kHz)	390.000zł
SOUND SAMPLER (stereo, 20kHz)	580.000zł
MIDI PRO (1*IN, 1*THRU, 2*OUT)	480.000zł
Rozszerzenie pamięci dla AMIGI 500 (512KB, bez zegara)	480.000zł
Program DIGITON V1.1	270.000zł
Elektroniczny BootSelektor DF0-DF3 dla A500/2000	320.000zł
KICK-BOX 2.0 dla A500	900.000zł

oraz wiele innych urządzeń

Ceny zawierają podatek VAT (stawka 22%)

Uwaga!!! - Sprzedaż również za zaliczeniem pocztowym po doliczeniu kosztów przesyłki.

Wszystkie urządzenia objęte są roczną gwarancją

Wszystkie urządzenia produkowane przez HDP Electronics można też zakupić w Gdańsku w firmie AMI-COMM, ul. Wały Jagiellońskie 1, tel. (058) 31-33-38, 31-29-84 w. 21.

ROZSZERZENIA PAMIĘCI

4 MB DO A 1200	4.700.000 !	
2 MB DO A 500	2.200.000	(TEST AMIGA 6/93)
2 MB DO A CDTV	2.200.000	
1 MB DO A 600	1.400.000	
1 MB DO A 500+	900.000	
0,5 MB DO A 500	450.000	

ELBOX RAM

sprzedaż również za zaliczeniem pocztowym po doliczeniu kosztów przesyłki

TEL./FAX: (0-12) 22-36-39
30-104 Kraków 45, skr. poczt. 99

ELBOX computer®



REAL v2.47

Marek Stor

Po raz pierwszy o nowej wersji Reala 2.4 usłyszałem na targach w Kolonii. Jak to na targach bywa nowa wersja była nie do kupienia, a panowie z firmy obiecywali, że w jeszcze przed nowym rokiem ukaże się w sprzedaży.

Na pytanie jak otrzymać nową wersję, (jestem szczęśliwym posiadaczem oryginału - tzw. update), odesłano mnie do przedstawiciela firmy Activa International w Polsce.

Po wielu telefonach wykonanych po powrocie do kraju, jednego czego udało mi się dowiedzieć u polskiego przedstawiciela to to, że wersja 2.4. będzie mnie kosztowała 129 DM.

Moja cierpliwość skończyła się wraz ze styczniem. Zadzwoniłem do Holandii gdzie miły pan poinformował mnie, że wersja Real'a 2.4. jest do kupienia już od połowy grudnia. Nie będę pisał co myślałem wtedy o polskim przedstawicielu Activa International. Wysłałem faksa z zamówieniem i po dwóch tygodniach miałem nową wersję u siebie w domu. Pomijam milczeniem fakt, że w dobie praw autorskich kazano mi dopłacić ponad 50% opłat celnych i podatkowych. To chyba żeby chronić nasz rynek od napływu imperialistycznej myśli naukowej.

Nowa wersja Reala jest dostarczona na czterech dyskietkach wraz z podręcznikiem składającym się z 58 stron opisujących nowe funkcje. Program posiada standartowy comodorowski instaler i nie ma problemu z zainstalowaniem nowej wersji obok starej. Zabezpieczenie nie zostało zmienione i obie wresje korzystają z tego samego dżongla wkładanego w port joystika.

Real 3D automatycznie rozpoznaje czy pracuje z procesorem 68020, 68030 czy 68040. Po uruchomieniu programu pokazuje się taki sam układ okien jak w wersji 2.3. Pierwsza zmiana jaka rzuca się w oczy dotyczy okna z narzędziami, gdzie w każdym rzęcie mamy funkcje pogrupowane w pojedyncze ikony i kliknięcie na tę ikonę powoduje włączanie lub wyłączanie wszystkich ikon, jakby w podkatalogu ikony głównej. Korzystanie z ikon w oknie z narzędziami stało się teraz o wiele wygodniejsze.

Druga ważna zmiana dotyczy kontrolowania kamery. Kamera jest prezentowana w wygodniejszy sposób, który pokazuje zwrot kamery i pole widzenia. Nie ma niestety miejsca, aby wymieniać wszytkie modyfikacje tej opcji, więc przytoczę tylko kilka np: nowy tryb sterowania ruchem i położeniem kamery, obsługa kamery z klawiatury, itp.

Ulepszono także pracę przy konstruowaniu obiektów. Dodano możliwość wycinania obiektów przy pomocy krzywych typu freeform, wyginania kolistego, które może być użyte przy wyginaniu prostej rury w pierścień, zakręcanie obiektów wokół własnej osi, nowe funkcje owijania, itd. Wbudowano także ładowanie fontów trówymiarowych co umożliwia, ustawienie całego napisu dokładnie w jednej linii.

Najważniejszą rzeczą jaka daje się zauważyć w nowej wersji to przyspieszenie czasu liczenia animacji przynajmniej dwukrotnie. Przyspieszono także czas odświeżania ekranu w edytorze, a w trybie environment renderowanie przyspieszono czterokrotnie. Podczas liczenia animacji ukazuje nam się zegar z czasem generowania jednego rysunku.

Poprawiono także zarządzanie pamięcią. Porównywałem dwa takie same projekty liczone w trybie normal na wersji 2.3 i 2.4 - starsza wersja nie chciała go policzyć z braku pamięci, na nowej żadnych problemów z tym nie było.

Producent napisał, że Real 3D v.2.4 posiada ponad sto (sic!) nowych funkcji w stosunku do wersji 2.3. Opisywanie ich wszystkich mija się z celem tego artykułu, mam tylko nadzieję, że w przyszłości, gdy skończy się kurs o Imagine, Amigowiec udostępni swoje łamy Realowi. Poniżej postaram się podsumować nowe funkcje, niewymienione w tekście.

Nowe funkcje:

- Nowe cechy materiałowe takie jak nakładanie fraktalne, sterownik koloru typu marmur, granit, sortowanie alfabetyczne, autoindeksacja powtarzających się nazw.

- Wbudowane oświetlenie stożkowe (reflektorowe), cylindryczne. Dodano nowe atrybuty źródła światła jak kontrola natężenia z odległością, kontrola natężenia snopu światła, jasności.

- Rozbudowane narzędzia do tworzenia złożonych układów punktów materialnych.

- Wypełnianie przestrzeni cząsteczkami.

- Nowy specjalny obiekt podstawowy szkieletu w celu lepszego sterowania i poprawienia wyglądu, tarcie połączeń umożliwiające naturalny ruch.

- Możliwa jest edycja punktów szkieletu przy użyciu metody inverse kinematics. Można na przykład animować szkielet przy użyciu kilku tych metod, z których każda oddziaływuje na wybrany punkt szkieletu.

- Dodano także nowe atrybuty obiektów:

a) "No shadows" usuwa cienie z powierzchni obiektu, co zwiększa szybkość renderowania.

b) "Bump shadows" generuje nierówne brzegi cieni na krawędziach powierzchni z wgnieceniami.

- Nowa funkcja automatycznego śledzenia punktu patrzenia. Na przykład przy animowaniu kamery używana jest korekcja względem płaszczyzny horyzontu, a nie korekcja typu "najkrótsza droga", jak to było w wersji 2.35. Eliminuje to przechylanie się kamery.

- Ulepszone sterowanie pomiędzy obiektem a kamerą. Można sterować skalą i kątem. Sterowanie i zerowanie współrzędnych punktu patrzenia oraz pozycji obserwatora, sterowanie odległością i głębią ostrości.

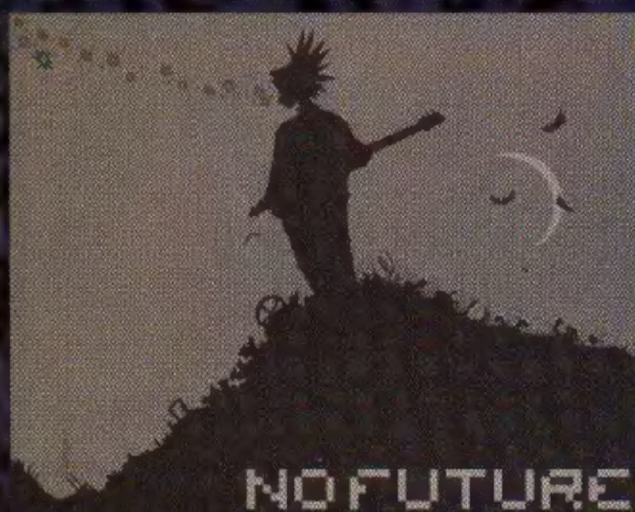
- RealConvert może teraz generować krzywe B-sklejane z plików ProDraw. Jest to szczególnie przydatne w połączeniu z nowym narzędziem do wyciskania obiektów typu freeform.

Mam nadzieję, że chociaż w małym stopniu przybliżyłem wam możliwości nowego Reala. Oczywiście nie jest on pozbawiony wad. Na przykład okno kontroli kamery mogło być trochę mniejsze. Problemy z okładaniem obiektów brushami, zwłaszcza jeżeli renderujemy obiekt w dużym zbliżeniu.

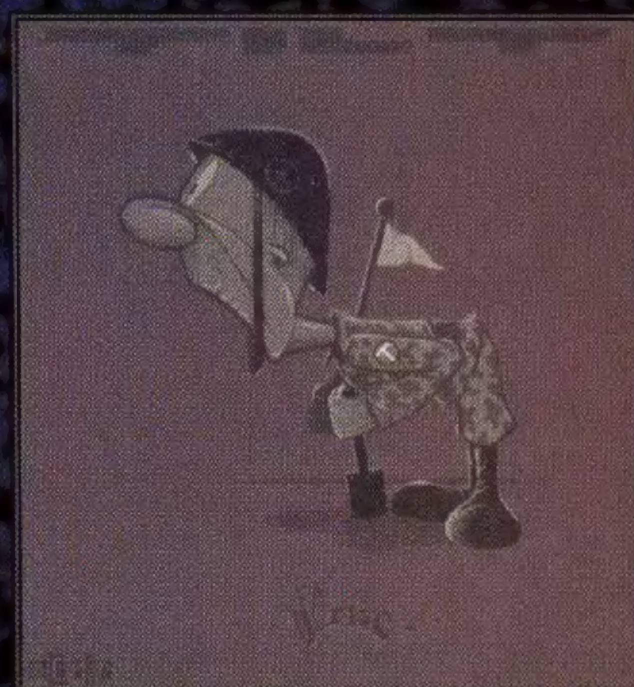
Podsumowując Real 3D wersja 2.4 jest w pełni profesjonalnym programem animacyjnym o dużych wymaganiach sprzętowych i powinna się pojawić na rynku zamiast wersji 2.3, która była chyba wersją wyłączenie do testowania programu na nas klientach.

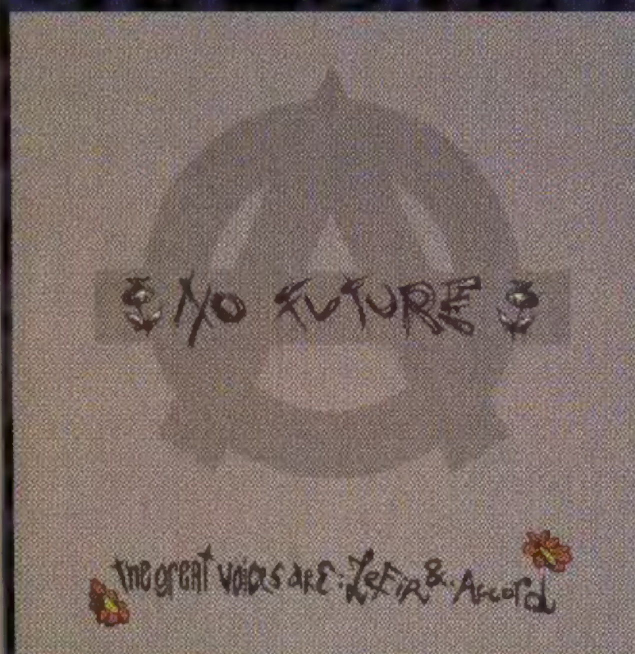
PS. Jeżeli coś zrobiłeś na Realu 3D v2.x, napisz. ❑





Voodoo Chile

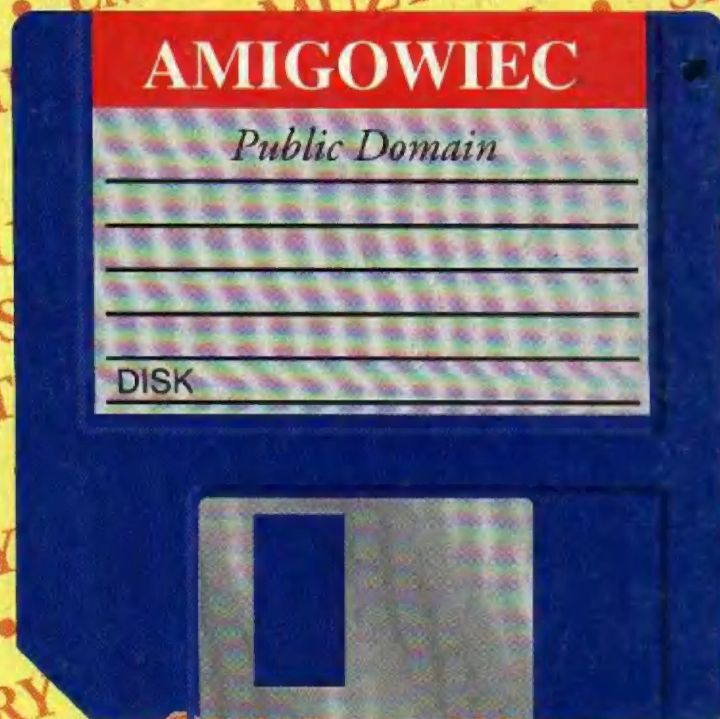








DYSKI PUBLIC DOMAIN



Nie używałeś nigdy programów Public Domain Amigowca? Spróbuj! 24 dyskietki czekają na Ciebie.

Zamawiając cały komplet, czyli 24 dyski zapłacisz o 30% taniej! 800.000 zł zamiast 1.150.000 zł! Zamówienia będą realizowane po wpłaceniu kwoty na nasze konto.

ALFIN

ul. Świętojańska 2/7
85-080 Bydgoszcz
tel. 28-79-20, fax 22-64-03
Konto: Bank PKO S.A. Bydgoszcz
5.09011-400933.9-2511-30-111.0